



Universidad de Cuenca

Facultad de Ingeniería

Escuela de Electrónica y Telecomunicaciones

Control de topología en redes inalámbricas de tipo malla con estructura de comunidades

*Trabajo de titulación previo a la
obtención del título de Ingeniero en
Electrónica y Telecomunicaciones.*

Autores :

Juan Diego Belesaca Mendieta C.I. 0104817770

Cristihan Ruben Criollo Cumbe C.I. 0103986295

Director :

Ing. Andrés Marcelo Vázquez Rodas, PhD C.I. 0301496840

Cuenca - Ecuador

2018



Resumen

Las redes inalámbricas de tipo malla ([WMNs](#)) son redes cuyo objetivo principal es el de brindar conectividad ubicua e inalámbrica a sus clientes a través de un conjunto de enrutadores conectados en malla ([MR](#)). La arquitectura de [WMNs](#) se pueden clasificar en 3 grupos dependiendo de las funcionalidades de los nodos: infraestructura, [WMNs](#) cliente, y las redes híbridas como una combinación de las anteriores.

Las [WMNs](#) cliente son redes cuya gestión debe ser lo más espontánea posible. Hoy en día, las redes espontáneas están emergiendo como un posible nuevo paradigma de comunicación, caracterizado por una fuerte naturaleza de auto-organización y auto-mantenimiento. El caso más común es el de redes formadas principalmente por dispositivos móviles transportados por los usuarios. En estas redes espontáneas los dispositivos cumplen tanto el rol de interfaz de usuario final, así como el de enrutador de tráfico de sus pares. Las redes espontáneas están formadas generalmente por dispositivos portátiles. Cuando estos dispositivos se asocian a personas que se mueven de un lugar a otro, entonces poseen el atributo de movilidad humana debido al comportamiento social. La mayoría de estas redes con tales patrones de movilidad presentan una estructura organizada en comunidades.

Debido al avance tecnológico y de manera incremental, los teléfonos móviles generan un área de interés para los investigadores, con el fin de aprovechar información en tiempo real, para beneficiar el cuidado personal y del medio ambiente. La combinación de dichos dispositivos inteligentes con las relaciones sociales son un motivo para introducir características sociales en las estrategias de diseño de redes. Debido al gran número de dispositivos que pueden existir en estas redes espontáneas, se presentan problemas como: sobrecarga de mensajes de enrutamiento, alta probabilidad de colisión de paquetes, interferencia, ineficiencia energética, etc. Surge, por tanto, la necesidad de implementar mecanismos de control de topología que tomen en cuenta la interacción social de los nodos.

Bajo este contexto, en este proyecto se proponen y se evalúan tres esquemas de control de

topología basados en métricas de centralidad en combinación con algoritmos de detección de comunidades, para redes inalámbricas en malla con estructura de comunidades. Cada uno de los esquemas evaluados utiliza un método distinto de selección de nodos que cumplirán la función de enrutadores y por ende conformarán el núcleo central de la red.

Al evaluar los métodos se concluye que el método denominado *Community-Aware Highest Betweenness Centrality Neighbor* (**C-A HC_BN**) es el que logra mejor conectividad de la red con un número reducido de nodos seleccionados para todos los modelos de movilidad considerados. Un análisis de la evolución de las topologías reducidas es llevado a cabo. En todos los casos se tiene una media menor al 50 % de nodos que se escogen para que cumplan las funciones de enrutador, y el número de veces que un nodo cambia de estado en promedio está entre los 22 a 41 segundos dependiendo del modelo de movilidad. El método propuesto es comparado con un mecanismo de control de topología de un trabajo previo basado en métricas de centralidad pero que no considera la estructura en comunidades. Los resultados de extensas simulaciones con tráfico en tiempo real (UDP) concluyen que el método propuesto mejora el desempeño de la red y la eficiencia energética.

Por último se propone un mecanismo de control de topología adicional basado en algoritmos de árboles de expansión para la selección de los enrutadores. Este método logra topologías más reducidas con respecto a **C-A HC_BN**. Sin embargo, el desempeño de la red con tráfico se ve perjudicado por el reducido número de enrutadores.

La evaluación de la propuesta se realiza mediante extensas simulaciones empleando el software **ns-3**, un simulador de redes de eventos discretos basado en C++ de licencia gratuita, altamente difundido y empleado por la comunidad científica. Las trazas de movilidad son obtenidas mediante BonnMotion y **SUMO**, herramientas de generación y análisis de escenarios de movilidad. La generación y visualización de grafos de la red son obtenidos mediante Gephi. Otros programas complementarios que son utilizados para el procesamiento de datos son Octave y Python.

Palabras clave : control de topología, malla, red, ns-3, detección, comunidades, centralidad, movilidad, árboles de expansión.



Abstract

Wireless mesh networks (WMNs) are networks whose main objective is to provide ubiquitous and wireless connectivity to their clients through a set of mesh routers (MR). The WMNs architecture can be classified into 3 groups depending on the functionalities of the nodes: infrastructure, client WMNs, and hybrid wireless mesh networks as a combination of the above.

The client WMNs are networks whose management should be as spontaneous as possible. Nowadays, spontaneous networks are emerging as a possible new paradigm of communication, characterized by a strong self-organization and self-maintenance nature. The most common cases are networks formed mainly by mobile devices carried by people. In such spontaneous networks, the devices fulfill both the role of the end user interface, as well as being the traffic router of their peers. Spontaneous networks are usually conformed by portable devices. When these devices are associated with people moving from one place to another, they acquire the human mobility attribute. Most of these networks with such mobility patterns present an organized community structure.

Due to technological advance, mobile phones increasingly generate an area of interest for researchers, in order to take advantage of real-time information, to benefit personal and environment care. The combination of such smart devices with social relationships is a reason to incorporate social characteristics into network design strategies. Due to the large number of devices that can exist in these spontaneous networks, there are problems such as: overhead, high probability of packet collision, interference, energy inefficiency, etc. Therefore, arises the need to implement topology control mechanisms that take into account the nodes's social interaction.

Under this context, this project proposes and evaluates three topology control schemes based on centrality metrics in combination with community detection algorithms, for community-structured WMNs. Each of the evaluated schemes uses a different method for routers selection which will form the network backbone. When evaluating the methods, it is concluded that the *Community-Aware Highest Betweenness Centrality Neighbor* (C-A HC_BN) method achieves the best network connectivity with a reduced number of selected routers for all the mobility models



under study. An analysis of the evolution of the reduced topologies is carried out. In all cases, there is an average of less than 50 % of nodes that are chosen to fulfill the functions of routers, and the number of times that a node changes its state on average goes from 22 to 41 seconds depending on the mobility model. [C-A HC_BN](#) is also compared to a previous topology control mechanism which doesn't consider community structure. The simulation results with real-time traffic (UDP) conclude that our method is better in terms of network performance and energy efficiency.

Finally, we propose an additional topology control mechanism based on the minimum spanning tree algorithm. This method manages to obtain smaller topologies than [C-A HC_BN](#). However, network performance with traffic load is affected by the reduced number of [MR](#) in this method.

The evaluation of the proposal is carried out by means of extensive simulations using the [ns-3](#) software, a free licensed discrete event network simulator based on C ++, highly diffused and used by the scientific community. The mobility traces are obtained by BonnMotion and [SUMO](#), which are tools for generation and mobility scenario analysis. Generation and visualization of network graphs are obtained with Gephi. Other software that are used for data processing are Octave and Python.

Keywords : topology control, mesh, network, ns-3, community, detection, centrality, mobility, spanning tree.



Índice general

Resumen	III
Abstract	V
Índice general	VII
Índice de figuras	XIII
Índice de tablas	XVII
Dedicatoria	XXIX
Agradecimientos	XXXIII
Abreviaciones y acrónimos	XXXV
1. Introducción	1
1.1. Identificación del Problema	2
1.2. Justificación	2
1.3. Alcance	3
1.4. Objetivos	4



1.4.1. Objetivo General	4
1.4.2. Objetivos específicos	4
2. Marco Teórico	5
2.1. Redes En Malla	6
2.1.1. Estándar IEEE 802.11s	6
2.1.1.1. Descubrimiento Mesh	7
2.1.1.2. Peer Management Protocol	7
2.1.1.3. Protocolo de enrutamiento	7
2.2. Control de Topología	8
2.3. Modelos de Movilidad	9
2.3.1. Caminata Lévy (Lévy Walks)	9
2.3.2. Pequeño mundo en movimiento (Small World in Motion SWIM)	10
2.3.3. Caminata humana de acción mínima auto-similar (Self-Similar Least-Action Human Walk SLAW)	12
2.3.4. SMOOTH	15
2.3.5. Área de desastre	16
2.3.6. Restricciones Geográficas	16
2.4. Detección de comunidades	17
2.5. Métricas de centralidad	19
3. Trabajos relacionados	23
4. Control de topología basado en métricas de centralidad y estructura comunitaria	27



4.1. Escenarios bajo consideración	28
4.1.1. Escenario SWIM	28
4.1.2. Escenario SLAW	29
4.1.3. Escenario SMOOTH	29
4.1.4. Escenario de área de desastre	30
4.1.5. Escenario de restricciones geográficas	33
4.1.6. Escenario RWM	34
4.2. Detección de comunidades	34
4.3. Selección de Enrutadores	36
4.3.1. k nodos más centrales en cada comunidad	37
4.3.2. Vecino con mayor valor de Intra-Centralidad de la misma comunidad (C-A $HIntraC_{BN}$)	38
4.3.3. Vecino con mayor valor de Centralidad Betweenness de la misma comunidad (C-A HC_{BN})	39
4.4. Análisis de topologías construidas	41
4.5. Conclusiones	46
5. Metodología	47
5.1. Generación de trazas de movilidad: Bonnmotion	48
5.2. Generación de trazas de movilidad: SUMO	48
5.3. Creación de la matriz de adyacencia	49
5.4. Detección de Comunidades	52
5.5. Selección de Enrutadores	54
5.6. Simulación de red con tráfico	54



5.6.1. Gestión de las simulaciones	56
5.7. Gráficos de topologías	58
6. Resultados	61
6.1. Impacto de la carga de tráfico, en redes con control de topología	64
6.1.1. Tasa media de mensajes para la administración del enrutamiento	67
6.1.2. Cantidad de paquetes de datos de unidifusión reenviados en la red	69
6.1.3. Paquetes recibidos con éxito	71
6.1.4. Total de paquetes reenviados por cada paquete recibido con éxito	73
6.1.5. Eficiencia de la red en términos de proporción de entrega de paquetes	73
6.1.6. Consumo Energético	75
6.1.7. Retardo de paquetes extremo-extremo	80
6.2. Conclusiones	87
7. Control de topología basado en árboles de expansión	89
7.1. Introducción	90
7.2. Control de topología basado en Método de Árboles de Expansión	90
7.3. Análisis de las topologías construidas	93
7.4. Impacto de la carga de tráfico en la red	97
7.5. Conclusiones	106
8. Conclusiones	107
8.1. Conclusiones	108
8.2. Limitaciones, Mejoras y Trabajos Futuros	109



A. Detección de comunidades	111
A.1. Number of communities detected	112
A.2. Community Detection modularity obtained	113
B. Scripts	115
B.1. Script principal de Octave (MAIN.m)	116
B.2. Script principal de ns-3 (MAIN.cc)	119
Bibliografía	127





Índice de figuras

2.1. Comportamiento modelo SWIM en base al valor de α	11
2.2. Movimientos de un nodo a lo largo del tiempo para el modelo de movilidad SLAW	14
4.1. Traza resultante del modelo de área de desastre.	31
4.2. Proceso de obtención de la traza con el escenario de restricciones geográficas.	33
4.3. Topología de la red para el escenario con modelo de movilidad SWIM en $t = 30seg$	36
4.4. Método <i>k nodos más centrales en cada comunidad</i> para distintos valores de porcentaje de “nodos puente”, para el escenario SWIM en $t = 30 seg$	38
4.5. Métodos C-A HIntraC_BN y C-A HC_BN para distintos valores de porcentaje de “nodos puente”, para el escenario SWIM en $t = 30 seg$	39
4.6. Topología reducida y backbone resultante con el método <i>k nodos más centrales en cada comunidad</i> con el 25 % de “nodos puente”, para el escenario con movilidad SWIM en $t = 30seg$	40
4.7. Topología reducida y backbone resultante con el método C-A HIntraC_BN seleccionando el 25 % de “nodos puente” para el escenario con movilidad SWIM en $t = 30seg$	40
4.8. Topología reducida y backbone resultante con el método C-A HC_BN seleccionando el 25 % de “nodos puente” para el escenario con movilidad SWIM en $t = 30seg$	40
4.9. Histogramas de cambios de estado de los nodos	44



4.10. Evolución del número de aristas en la red a través de 500 segundos de simulación. Resultados para los seis distintos escenarios.	45
5.1. Comportamiento de los 4 núcleos del computador al correr simulaciones	57
5.2. Pajek file example output	59
6.1. Resultados obtenidos de <i>All Routers</i> en comparación con los métodos de control de topología $2HC_BN$ y $C-A HC_BN$, para el escenario con modelo de movilidad $SWIM$	65
6.2. Resultados de la tasa de mensajes para gestión de enrutamiento en función del número de flujos para cada modelo de movilidad.	68
6.3. Topología generada con 14 nodos en malla	69
6.4. Resultados del total de paquetes unidifusión reenviados en función del numero de flujos para cada modelo de movilidad.	70
6.5. Resultados del total de paquetes recibidos con éxito en función de la cantidad de flujos simulatáneos para cada modelo de movilidad.	72
6.6. Resultados del total de paquetes reenviados por cada paquete recibido con éxito en función de la cantidad de flujos activos para cada modelo de movilidad.	74
6.7. Resultados de la eficiencia de la red en términos de la tasa de paquetes entregados en función de la cantidad de flujos activos para cada modelo de movilidad.	76
6.8. Histograma de consumo de energía de los nodos para el caso de 20 flujos activos. Resultados para cada modelo de movilidad.	78
6.9. Función de distribución acumulada del consumo de energía por nodo para el caso de 20 flujos activos. Resultados para cada modelo de movilidad.	79
6.10. Función de distribución acumulada del retardo de paquetes extremo-extremo para el modelo $SWIM$	81
6.11. Función de distribución acumulada del retardo de paquetes extremo a extremo para el modelo $SLAW$	82



6.12. Función de distribución acumulada del retardo de paquetes extremo a extremo para el modelo SMOOTH.	83
6.13. Función de distribución acumulada del retardo de paquetes extremo a extremo para el modelo de área de desastre.	84
6.14. Función de distribución acumulada del retardo de paquetes extremo a extremo para el modelo con restricciones geográficas.	85
6.15. Función de distribución acumulada del retardo de paquetes extremo a extremo para el modelo RWM.	86
7.1. Topologías creadas con el método de árbol de expansión (ST) para el modelo SWIM en $t = 30seg$	92
7.2. Histogramas de la cantidad de cambios de estado de los nodos, para los métodos C-A HC _B N y ST para el tiempo total de simulación.	95
7.3. Evolución en el tiempo del número de aristas en la red a través de los 500 segundos de duración de cada escenario, para los métodos C-A HC _B N y ST	96
7.4. Resultados de la tasa de mensajes para gestión de enrutamiento en función del número de flujo para cada modelo de movilidad.	98
7.5. Resultados del total de datos reenviados en función del numero de flujo para cada modelo de movilidad.	99
7.6. Resultados del total de paquetes recibidos con éxito en función del numero de flujo para cada modelo de movilidad.	100
7.7. Resultados del total de datos reenviados por paquete recibido con éxito en función del numero de flujo para cada modelo de movilidad.	101
7.8. Resultados de la eficiencia de la red en términos PDR en función del numero de flujo para cada modelo de movilidad.	102
7.9. Histograma de consumo de energía por nodo para el caso de 20 flujos, resultados para cada modelo de movilidad.	103
7.10. Función de distribución acumulada del consumo de energía por nodo para el caso de 20 flujos, resultados para cada modelo de movilidad.	104



A.1. Number of communities detected	112
A.2. Community Detection modularity obtained	113



Índice de tablas

4.1. Parámetros de entrada para SWIM	29
4.2. Parámetros de entrada para SLAW	29
4.3. Parámetros de entrada para SMOOTH.	30
4.4. Parámetros de entrada para Área de Desastre.	31
4.5. Disposición de sub áreas estratégicas para el área de desastre.	32
4.6. Modularidades alcanzadas con detección de comunidades	35
4.7. Número de enrutadores seleccionados para valores mínimos de $F = 0$	41
4.8. Número de enrutadores seleccionados con el método C-A HC_BN para escenarios de 500 segundos	42
4.9. Número de cambios de estado de los nodos con el método C-A HC_BN para escenarios de 500 segundos	43
4.10. Número promedio de aristas en la topología durante 500 segundos de simulación.	43
4.11. Porcentaje de disminución de aristas en la topología aplicando el método C-A HC_BN con respecto a topología completa durante 500 segundos de simulación.	43
5.1. Comparativa de formatos soportados por Gephi	58
6.1. Parámetros RF de los nodos	63



6.2. Resumen de porcentajes promedio de mejora de los mecanismos de control $2HC_BN$ y $C-A HC_BN$ con respecto a la red operando con todos los nodos como enrutadores, para cada parámetros y modelo de movilidad.	66
6.3. Porcentajes promedios de reducción de mensajes de enrutamiento con el método $C-A HC_BN$ con respecto a $2HC_BN$	67
6.4. Porcentaje de reducción del total de paquetes reenviados por la red con el método $C-A HC_BN$ con respecto a $2HC_BN$	71
6.5. Porcentaje de incremento de paquetes recibidos con éxito con el método $C-A HC_BN$ con respecto a $2HC_BN$	71
6.6. Porcentaje de disminución de datos reenviados por paquete recibido con éxito con el método $C-A HC_BN$ con respecto a $2HC_BN$	73
6.7. Porcentaje de incremento de la eficiencia de la red en términos de proporción de entrega de paquetes, del método $C-A HC_BN$ con respecto a $2HC_BN$	75
6.8. Consumo energético promedio de toda la red para 20 flujos activos, con el método $2HC_BN$ y $C-A HC_BN$	77
6.9. Porcentaje de disminución de consumo energético de toda la red para el caso de 20 flujos activos, del método $C-A HC_BN$ con respecto al método $2HC_BN$	80
7.1. Promedio de enrutadores seleccionados para los métodos $C-A HC_BN$ vs ST en escenarios de 100 nodos y 500 segundos de duración	93
7.2. Promedio de cambios de estado de los nodos para los métodos $C-A HC_BN$ vs ST en escenarios de 100 nodos, 500 segundos de duración, y diferentes modelos de movilidad	94
7.3. Resumen de porcentajes promedios de mejora del método ST con respecto a <i>todos enrutadores</i> y la diferencia con el método propuesto $C-A HC_BN$	105



Cláusula de Propiedad Intelectual

Yo Cristihan Ruben Criollo Cumbe, autor del trabajo de titulación “Control de Topología en Redes Inalámbricas de tipo Malla con Estructura de Comunidades”, certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 9 de abril de 2018

Cristihan Ruben Criollo Cumbe

C.I: 0103986295





Cláusula de Propiedad Intelectual

Yo Juan Diego Belesaca Mendieta, autor del trabajo de titulación “Control de Topología en Redes Inalámbricas de tipo Malla con Estructura de Comunidades”, certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 9 de abril de 2018

Juan Diego Belesaca Mendieta

C.I: 0104817770





Cláusula de licencia y autorización para publicación en el Repositorio Institucional

Yo Cristihan Ruben Criollo Cumbe en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación “Control de Topología en Redes Inalámbricas de tipo Malla con Estructura de Comunidades”, de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos.

Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, 9 de abril de 2018

Cristihan Ruben Criollo Cumbe

C.I: 0103986295





Cláusula de licencia y autorización para publicación en el Repositorio Institucional

Yo, Juan Diego Belesaca Mendieta en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación “Control de Topología Topología en Redes Inalámbricas de tipo Malla con Estructura de Comunidades”, de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos.

Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, 9 de abril de 2018

Juan Diego Belesaca Mendieta

C.I: 0104817770





Certifico

Que el presente trabajo de titulación: “Control de Topología en Redes Inalámbricas de tipo Malla con Estructura de Comunidades”, fue dirigido y revisado por mi persona.

A handwritten signature in blue ink, reading "VÁZQUEZ", with a large, stylized flourish above it.

Ing. Andrés Marcelo Vázquez Rodas, PhD

Director





Dedicatoria

A mis padres,

por ser un pilar fundamental en mi formación académica y personal, por sus consejos y valores inculcados que me hicieron crecer como persona, por su constante apoyo y motivación que me permitió concluir esta etapa tan importante de mi vida. Todo este trabajo ha sido posible gracias a ellos.

A mi hermana,

por motivarme a mejorar día con día, por ser una amiga y confidente, te quiero mucho.

A mis amigos,

que gracias a su ayuda logramos una formación profesional conjunta, y todos aquellos supieron compartir buenos y malos momentos que marcaron mi vida.

Finalmente a los maestros, aquellos que supieron guiar mi camino universitario, compartiendo sus conocimientos, exigencias y formando un carácter profesional en mí.

Juan Diego Belesaca





Dedicatoria

A Rubén, Nube y Bryan,

quienes son y serán las personas más importantes en mi vida. Cada logro y cada meta cumplida se las dedico a ustedes. Gracias por educarme y formarme en la persona que soy el día de hoy. Espero llenarles de orgullo siempre y nunca fallarles. Les quiero mucho familia.

Cristihan Criollo





Agradecimientos

A nuestro director Andrés Vázquez,

por confiar en nosotros para llevar acabo este trabajo. Gracias por su esfuerzo, dedicación, tutoría y motivación que han sido imprescindibles para el desarrollo de este trabajo.

Al Ingeniero Fabián Astudillo,

que nos brindó ayuda incondicional, con sus amplios conocimientos de ns-3 y sistemas operativos.

Al Ingeniero Lisandro Solano y a la Universidad de Cuenca,

por permitirnos hacer uso de los equipos que hicieron posible llevar a cabo las innumerables simulaciones de este proyecto.

¡Gracias a ustedes!

Juan Diego Belesaca, Cristihan Criollo





Abreviaciones y Acrónimos

2HC_BN 2 Highest Betweenness Centrality Neighbors. 62, 64–67, 69, 71, 73, 75, 77, 80, 87, 109

AODV Ad hoc On Demand Distance Vector. 7

AP Access Point. 6

BSS Basic Service Set. 6

C-A HC_BN Community-Aware Highest Betweenness Centrality Neighbor. 37, 39–43, 46, 54, 62, 64–67, 69, 71, 73, 75, 77, 80, 87, 90, 93–97, 105, 106, 108, 109

C-A HIntraC_BN Community-Aware Highest Betweenness Intra-Centrality Neighbor. 37–40

CCS Casualties Clearing Station. 30, 71

CDS Connected Dominating Set. 9, 25, 26, 62

DS Dominating Set. 9

DTN Delay Tolerant Network. 24, 25

FTMRT Fault-Tolerant Multi-Routing Tree. 24

HWMP Hybrid Wireless Mesh Protocol. 7, 50, 55, 62

HWSN Heterogeneous Wireless Sensor Network. 24

IBSS Independent Basic Service Set. 6

ICT Inter-Contact Times. 10–12

IL Incident Location. 30

IoT Internet of Things. 2, 24

MAC Medium Access Control. 6

MC Mesh Client. 39, 42, 90, 93

MR Mesh Router. 2, 37, 39–42, 46, 62, 64, 67, 89, 90, 93, 94, 97, 106, 109

MSLAW Map-Based Self-similar Least Action Human Wlak. 17

ns-3 Network Simulator 3. 4, 49, 54–56, 62



OSM Open Street Map. [17](#), [33](#), [48](#), [49](#)

P2P Peer-to-Peer. [7](#), [62](#)

PDR Packet Delivert Ratio. [62](#), [73](#), [97](#)

PREQ Path Request. [7](#), [8](#), [55](#)

PSO Particle Swarm Optimization. [24](#)

PWT Patients Witing for Treatment area. [30](#)

RF Radio Frequency. [63](#)

RWM Random Waypoint Movement. [2](#), [9](#), [12](#), [28](#), [34](#), [42](#), [46](#), [48](#), [69](#), [71](#), [73](#), [86](#)

SLAW Self-similar Least Action Walk. [3](#), [10](#), [12–16](#), [28](#), [29](#), [46](#), [48](#), [69](#), [71](#), [73](#), [77](#), [80](#), [82](#)

SNA Social Netwrok Analysis. [25](#)

ST Spanning Tree. [90](#), [92–97](#), [106](#), [109](#)

STA Station. [6](#)

SUMO Simulation of Urban MObility. [4](#), [28](#), [33](#), [48](#), [49](#), [55](#)

SWIM Small World In Motion. [3](#), [10](#), [11](#), [16](#), [28](#), [29](#), [35–42](#), [46](#), [48](#), [64](#), [65](#), [67](#), [69](#), [71](#), [73](#), [81](#), [91–93](#), [106](#)

TDMA Time Division Multiple Access. [25](#)

TLW Truncated Lévy Walks. [10](#), [12](#)

TOC Technical Operational Command. [30](#)

UDP User Datagram Protocol. [55](#), [61](#), [62](#), [87](#)

WLAN Wireless Local Area Network. [6](#)

WMN Wireless Mesh Network. [2](#), [5](#), [6](#), [8](#), [9](#), [24](#), [25](#), [62](#), [64](#), [90](#), [109](#)

WSN Wireless Sensor Network. [8](#), [24](#)



Capítulo 1

Introducción

En este capítulo presentamos la identificación del problema y justificación del proyecto a realizar, así como el alcance y objetivos del mismo.



1.1. Identificación del Problema

Las redes inalámbricas de tipo malla (**WMNs**) son redes de múltiple salto auto-configurables de bajo costo cuyo objetivo principal es el de brindar conectividad ubicua e inalámbrica a sus clientes a través de un conjunto de enrutadores en malla (Mesh Routers, **MRs**). Los **MRs** son típicamente equipos poderosos con varias interfaces de radio que crean un backbone robusto para proporcionar conectividad a los nodos [1]. La arquitectura de **WMNs** se puede clasificar en 3 grupos dependiendo de las funcionalidades de los nodos: infraestructura (los enrutadores mesh forman una infraestructura para clientes tradicionales que se conectan a ella), **WMNs** cliente (forman, generalmente de manera ad-hoc, redes peer-to-peer entre los dispositivos de usuario final), y las redes híbridas como una combinación de las anteriores [2].

Las **WMNs** cliente son redes cuya gestión debe ser lo más espontánea como sea posible. Hoy en día, las redes espontáneas [3] están emergiendo como un posible nuevo paradigma de comunicación, caracterizado por una fuerte naturaleza de auto-organización y mantenimiento. El caso más común es el de redes formadas principalmente por dispositivos móviles transportadas por los usuarios. Con los avances actuales estas redes podrían convertirse en sistemas de sensado completo dentro del paradigma de Internet de las cosas (**IoT**), con una innumerable cantidad de posibles aplicaciones y campos de acción [4]. Por tanto, cada vez más los teléfonos móviles generan un área de interés para los investigadores con el fin de aprovechar la información en tiempo real y beneficiar el cuidado personal y del medio ambiente.

En tales redes espontáneas, los dispositivos cumplen tanto el rol de interfaz de usuario final, así como el de ser enrutador de tráfico de sus pares [3]. Sin embargo, debido al gran número de dispositivos que pueden existir en estas redes espontáneas, no es conveniente ni eficiente que todos los nodos cumplan el rol de enrutadores. Generalmente se presentan problemas como: sobrecarga de mensajes de enrutamiento, alta probabilidad de colisión de paquetes, interferencia, ineficiencia energética, etc. [5].

1.2. Justificación

Como se ha mencionado anteriormente, las redes espontáneas están formadas generalmente por dispositivos portátiles (principalmente teléfonos móviles). Cuando estos dispositivos se asocian a personas que se mueven de un lugar a otro, entonces éstos dispositivos adquieren un patrón de movilidad humana. Hace algunos años los modelos de movilidad utilizados por la mayoría de investigadores en redes ad-hoc eran aleatorios, como por ejemplo el modelo de movilidad aleatoria (Random Waypoint Mobility **RWM**) [6]. La principal ventaja de este tipo de modelos de movilidad era su sencillez; sin embargo, las trazas obtenidas no se asemejan a la realidad.

Posteriormente, la comunidad científica determinó que para modelar apropiadamente el movimiento humano se deben tomar en cuenta 3 propiedades principales: espaciales, temporales, y sociales [7]. Llegándose a considerar incluso que las más importantes son aquellas propiedades basadas en el comportamiento social y son estas las que definen las características de conectividad entre distintos usuarios. En este contexto, han surgido nuevos modelos de movilidad, que en base a contrastación con trazas reales, se aproximan de mejor manera al comportamiento real del movimiento de los seres humanos. Entre estos modelos se pueden citar ejemplos tales como [SWIM](#) [8], [SLAW](#) [9], [SMOOTH](#) [10], Área de Desastre [11], entre otros.

Como consecuencia de esta asociación humana, la mayoría de estas redes resultantes con tales patrones de movilidad, presentan adicionalmente una estructura organizada en comunidades. En el contexto de la teoría de grafos, una comunidad es considerada como un grupo de nodos con mejor interacción entre sus miembros, que con el resto de miembros de la red [12, 13]. Para la detección de comunidades existen varios algoritmos que se han trabajado a lo largo de los años, estos principalmente están basados en: aproximaciones para encontrar conglomeraciones de nodos [14, 15]; heurísticas [16, 17], heurísticas de coloreado balanceado [18]; optimización [19, 20] y otros como Girvan y Newman [13] presentan un método para encontrar comunidades, y límites de estos mediante índices de centralidad (i.e, betweenness centrality) ya sea en los nodos o en los enlaces entre nodos. Varios de estos trabajos se sintetizan en [21, 22].

Entonces, la combinación de los numerosos dispositivos inteligentes con las relaciones sociales inter-personales son un motivo para introducir características sociales en las estrategias de diseño de redes [23]. Debido a los problemas que se presentan en redes espontáneas con numerosos dispositivos, surge la necesidad de implementar mecanismos de control de topología que tomen en cuenta la interacción social de los nodos.

1.3. Alcance

El alcance de este proyecto es el de diseñar, implementar y evaluar un mecanismo de control de topología que tome en cuenta métricas de centralidad y la estructura comunitaria de una red en malla. Para ello, primero se pretende estudiar y evaluar distintos modelos de movilidad, para posteriormente implementarlos en los nodos de la red en malla. Una vez que los nodos estén en movimiento, se procederá a diseñar un control de topología basado en una combinación de métricas de centralidad y esquemas de detección de comunidades. El resultado final será un procedimiento de selección de los nodos que cumplan la función de enrutador. Se realizará una comparación de los resultados obtenidos, con aquellos de una red sin control de topología y con un control de topología que no considere estructuras comunitarias, para distintos posibles escenarios. Adicionalmente se compararán las topologías reducidas resultantes de esta propuesta

con esquemas de control de topología basados en árboles de expansión.

La evaluación de la propuesta se realiza mediante simulaciones empleando el software [ns-3](#) [24], un simulador de redes de eventos discretos basado en C++ de licencia gratuita, altamente difundido y empleado por la comunidad científica. Las trazas de movilidad son obtenidas mediante BonnMotion [25] y SUMO [26], herramientas de generación y análisis de escenarios de movilidad. La generación y visualización de grafos de la red serán obtenidos mediante Gephi [27]. Otros programas complementarios que serán utilizados para el procesamiento de datos son Octave [28] y Python [29].

1.4. Objetivos

1.4.1. Objetivo General

- Evaluar esquemas de control de topología basados en métricas de centralidad y algoritmos de detección de comunidades para redes inalámbricas en malla con estructura de comunidades.

1.4.2. Objetivos específicos

- Evaluar distintos modelos de movilidad basados en comportamiento social.
- Evaluar mecanismos de control de topología basados en métricas de centralidad.
- Complementar los mecanismos de control de topología con detección de comunidades.
- Evaluar y comparar el esquema de control de topología combinado con algoritmos basados en árboles de expansión.



Capítulo 2

Marco Teórico

En este capítulo describimos los principales protocolos que involucran a las [WMNs](#), hablamos sobre los distintos modelos de movilidad que utilizaremos, presentamos el concepto de detección de comunidades, y finalmente repasamos algunas métricas de centralidad que serán útiles para el presente trabajo.



2.1. Redes En Malla

Las redes inalámbricas de tipo malla (**WMNs**) son redes multi-salto, auto-configurables, de bajo costo, cuyo objetivo principal es el de brindar conectividad ubicua e inalámbrica a sus clientes a través de un conjunto de enrutadores.

La arquitectura de las **WMNs** se puede clasificar en 3 grupos principales en base a la funcionalidad de sus nodos [2]. En primer lugar están las **WMNs Infraestructura** en las cuales los enrutadores forman una infraestructura o *backbone* en malla para brindar conectividad a los clientes tradicionales que se conectan a ella. Permite la integración de las **WMNs** con otros tipos de redes mediante enrutadores con funcionalidades de *Gateway* o *Bridge*. En segundo lugar están las **WMNs Cliente**, que forman de manera ad-hoc, forman redes peer-to-peer entre los dispositivos de usuario final. En este caso, son los mismos dispositivos de usuario final los que desempeñan las tareas de enrutamiento y configuración. Finalmente, las **WMNs híbridas** son una combinación de las anteriores.

2.1.1. Estándar IEEE 802.11s

El estándar IEEE 802.11 [30] especifica las capas física, **MAC**, y de enlace para las redes inalámbricas de área local **WLAN**. En el estándar se definen dos tipos de **WLAN**. La primera basada en infraestructura (**BSS**), consiste de un punto de acceso (**AP**) y un número de estaciones asociadas (**STA**). La segunda sin infraestructura (**IBSS**), es un modo ad-hoc. En las redes ad-hoc [31], los usuarios se pueden comunicar entre ellos sin la necesidad de una infraestructura fija. El estándar 802.11 provee topologías de redes con uno o dos saltos [32]. Sin embargo, debido a que el consumo energético de los dispositivos es directamente proporcional a la distancia entre ellos, las transmisiones de uno o dos saltos podrían requerir mayores potencias de transmisión, ocasionando interferencia en la red e ineficiencia energética.

El grupo de trabajo 802.11s [33] es una enmienda del estándar que ha sido incluido en el IEEE 802.11-2012 que permite el reenvío de tramas de forma transparente en topologías arbitrarias de varios saltos. La principal característica es que define la selección de rutas y mecanismos de reenvío en la capa de enlace en lugar del tradicional enrutamiento de capa 3. Cada estación *mesh* que participa en una red inalámbrica opera como un enrutador de capa de enlace y es responsable de cooperar con todas las demás estaciones en la entrega de la trama. Esto se logra gracias a varios protocolos dedicados y funcionalidades. Los más importantes definidos en el estándar son Descubrimiento Mesh, Protocolo de Administración de Pares (Peer Management Protocol) y el Protocolo de Enrutamiento, que se detallan a continuación.



2.1.1.1. Descubrimiento Mesh

Consiste en un proceso de escaneo que provee la información de perfil de una estación. Este perfil contiene atributos como el identificador *mesh* (mesh ID), el protocolo de selección de rutas, la métrica de selección de rutas, el modo de control de congestión, el método de sincronización y el protocolo de autenticación. Todos los nodos dentro de una red en malla deben tener el mismo perfil.

2.1.1.2. Peer Management Protocol

Es el protocolo utilizado para el descubrimiento y mantenimiento de la topología de red. Este protocolo utiliza sus propias tramas para abrir, administrar y cerrar enlaces entre estaciones vecinas en una red mesh. Para detectar vecinos (mesh discovery), cada estación mesh envía periódicamente tramas de administración de un salto conocidas como *beacons*. Un enlace entre pares se establece solamente si las dos estaciones enviaron peticiones *Peering Open* y recibieron exitosamente respuestas de confirmación *Peering Confirm* cada una. El estándar no especifica cuándo se debe cerrar un enlace entre pares. Sin embargo a lo largo de este trabajo se utiliza un valor de pérdidas consecutivas de beacons, o transmisiones de datos fallidas para detectar cuando un enlace se rompe.

2.1.1.3. Protocolo de enrutamiento

Es la funcionalidad principal que provee capacidades de comunicación multi-salto. Como es usual, el protocolo de enrutamiento utiliza métricas de selección de rutas para escoger la trayectoria óptima de entre todas las posibles. El estándar IEEE incluye un protocolo de enrutamiento por defecto (Hybrid Wireless Mesh Protocol) y una métrica de selección de trayectoria por defecto (Airtime Link Metric).

Hybrid Wireless Mesh Protocol (HWMP) es un protocolo de enrutamiento basado en AODV y enrutamiento basado en árboles. HWMP soporta dos modos de operación dependiendo de su configuración. El modo **reactivo** o bajo demanda, consiste en descubrimiento de trayectorias y empieza cuando una estación fuente requiere transmitir datos a un destino desconocido. El nodo fuente difunde un mensaje de petición de ruta (PREQ), y los nodos intermedios crean/actualizan la ruta hacia el nodo destino. En el caso que un nodo intermedio no tiene una ruta hacia el destino, reenvía el mensaje PREQ a sus vecinos. Los protocolos de enrutamiento reactivos son más adecuados en redes con baja densidad de nodos y patrones de tráfico estático. Son los preferidos en redes con alta movilidad y comunicaciones entre pares P2P [34]. En el modo **proactivo**, se elige a una estación como nodo raíz del árbol de ruta y es el que difunde

mensajes **PREQ** de manera periódica. Para ello, todas las estaciones deben conocer el camino hacia la raíz y la raíz debe conocer las rutas hacia cada estación. Generalmente, los protocolos de enrutamiento proactivos se prefieren en redes pequeñas y con topologías fijas [34].

La métrica de selección de ruta multi-salto Airtime Link Metric mide la cantidad de recursos de canal consumidas al transmitir la trama sobre un enlace en particular. Se define como la suma de todas las métricas de enlace de tiempo-en-aire. Se asume que los mejores enlaces tienen menores métricas. Toma en cuenta el número total de reintentos, la tasa de datos actual, y algunas sobrecargas de acceso al medio. La métrica Airtime Link c_a se calcula con la siguiente ecuación [33]:

$$c_a = \left[O + \frac{B_t}{r} \right] \frac{1}{1 - e_f} \quad (2.1)$$

Donde O es el sobre-encabezado de acceso al canal (encabezados de trama, secuencias de entrenamiento, tramas de acceso al protocolo, etc.) que depende del tipo de canal Físico. B_t es el tamaño de la trama de prueba (se recomienda 8192 bits). r es la tasa de datos en Mbps a la cual una estación mesh transmitiría la trama de prueba y e_f es la tasa de error de trama medida para la trama de prueba.

2.2. Control de Topología

Control de topología se define como una técnica que utiliza cualquier parámetro de red para generar y mantener una topología, con el fin alcanzar una propiedad deseada en un red entera [5]. Control de topología es una de las técnicas más importantes utilizadas en redes ad hoc y WSNs cuya finalidad es principalmente la reducción de consumo energético e interferencia de radio (y por tanto colisión de paquetes) [35].

Una tarea importante de una red ad hoc conformada de nodos geográficamente dispersos como es el caso de las WMNs Cliente, es determinar una topología apropiada sobre la cual se implementarán protocolos de enrutamiento de alto nivel [36]. Estos protocolos de enrutamiento aprovecharán las características de la topología. Debido a que las topologías se encuentran cambiando constantemente, el esquema de enrutamiento debe ser robusto ante estos cambios.

La mayor parte de mecanismos de control de topología se basan en el **ajuste de potencia** de transmisión de sus nodos [35, 37–39]. Con niveles de potencia muy altos, se producen problemas de interferencia. Por el otro lado, niveles bajos de potencia resultan en una red desconectada

o fragmentada. Otros mecanismos se basan en los **estados de operación** de un nodo. Estos modos pueden ser principalmente los siguientes: *transmisión*, *recepción*, *idle*, y *sleep*. Finalmente, existen mecanismos que se enfocan en **agrupamiento**, generalmente mediante la selección estratégica del set dominante conectado (**CDS**) como en [40]. El problema de hallar un set dominante (**DS**) consiste en encontrar un subconjunto D de nodos en el grafo tal que cada nodo pertenezca a D o sea adyacente a D [41]. Si los nodos de este **DS** forman un grafo conectado, entonces el set es un **CDS**. Este conjunto de nodos pertenecientes al **CDS** formarán el *backbone* de la red.

La calidad de una topología resultante se puede medir de acuerdo a varios aspectos, como por ejemplo: conectividad, eficiencia energética, throughput, y robustez ante movilidad [36].

2.3. Modelos de Movilidad

En simulaciones de **WMNs** es importante emplear escenarios representativos lo más realistas posibles. Un registro de movimientos reales en un escenario y entorno específico es complicado de obtener debido al tiempo necesario y personal requerido, por lo que se opta por tener trazas sintéticas representadas por modelos de movilidad. Un modelo de movilidad representa mediante formulación matemática, el comportamiento humano. Por tanto, utilizar trazas sintéticas beneficia a la obtención de resultados de desempeño de red que se asemejan a la realidad.

Hace algunos años el modelo de movilidad utilizado por la mayoría de investigadores era el modelo de punto de referencia aleatorio (Random Waypoint Mobility **RWM**) [6]. Su principal ventaja es su sencillez; sin embargo las trazas obtenidas no se asemejan a la realidad. A continuación se describen algunos modelos de movilidad que toman en cuenta un mayor número de características del comportamiento humano y que se han considerado como los mas relevantes para el presente estudio.

2.3.1. Caminata Lévy (Lévy Walks)

La idea principal de este modelo es que el comportamiento humano puede ser representado por ciertas características estadísticas como movimientos *Lévy Walks* [42]. Esto es, un conjunto de movimientos aleatorios gobernados por difusión. Inicialmente fue utilizado para representar el comportamiento de algunas especies animales [43, 44]. Posteriormente, Rhee et al., en [45] mostró que el modelo puede acotarse con el objetivo de asemejar decisiones humanas para trasladarse hacia su destino, dando lugar así al modelo caminata de Lévy truncada (Truncated

Lévy Walks [TLW](#)). Este modelo de movilidad ha sido tomado como punto de partida para el desarrollo de distintos modelos (e.g., [SWIM](#), [SLAW](#), [SMOOTH](#)).

2.3.2. Pequeño mundo en movimiento (Small World in Motion [SWIM](#))

[SWIM](#) [8], es un modelo de movilidad muy eficiente y simple de implementar. No requiere demasiados parámetros de entrada. Se basa principalmente en capturar el comportamiento social de las personas como: el frecuentar lugares cercanos al hogar o frecuentar sitios donde pueden socializar con otras personas. Esto da paso a la característica principal de este modelo que es la distribución de tiempo de contacto entre nodos (Inter-Contact Times [ICT](#)) . El tiempo beneficia las oportunidades para establecer comunicación entre dispositivos, ampliando su uso en redes móviles ad-hoc. La distribución de tiempo es tomada de experimentos realizados en [46, 47], donde el primer intervalo es descrito como una ley de potencias y luego es truncado por una distribución exponencial cortada, mejor conocida como la dicotomía de decaimiento exponencial.

Los principales parámetros de entrada del modelo [SWIM](#) son:

- Peso de la distancia de celda α
- Exponente de tiempo de espera
- Límite superior del tiempo de espera
- Rango de transmisión de los nodos
- Multiplicador de la velocidad del nodo

Este modelo describe el movimiento de los nodos como se explica a continuación. El área de trabajo es dividida en celdas rectangulares de igual tamaño al rango de transmisión de los nodos (Fig. 2.1). A cada nodo se le asigna un peso que se utiliza para la planificación de próximos destinos. Este valor aumenta dependiendo de la popularidad (conglomeración de nodos) de la celda en el que se encuentre, y disminuye si se encuentra alejado de su hogar (el hogar se asigna a cada nodo aleatoriamente dentro del área). El cálculo del próximo destino se realiza mediante el peso de la distancia de celda α , que toma valores entre 0 y 1. El valor de α influye en el comportamiento de los nodos de la siguiente manera: mientras más alto el valor de α , el nodo tiende a visitar lugares cercanos a su hogar (Fig. 2.1a). Entonces los nodos tienen una relación social con sus vecinos cercanos que frecuentan los mismos lugares. En cambio, si el valor de α es bajo los nodos tienden a visitar lugares con mayor conglomeración de nodos en toda el área de trabajo (Fig. 2.1b), permitiendo socializar con mayor número de individuos. El nodo se traslada hacia su destino a una velocidad constante, emulando una caminata humana. El valor de velocidad se establece ecuánime a la distancia de recorrido entre dos puntos. Adicionalmente,

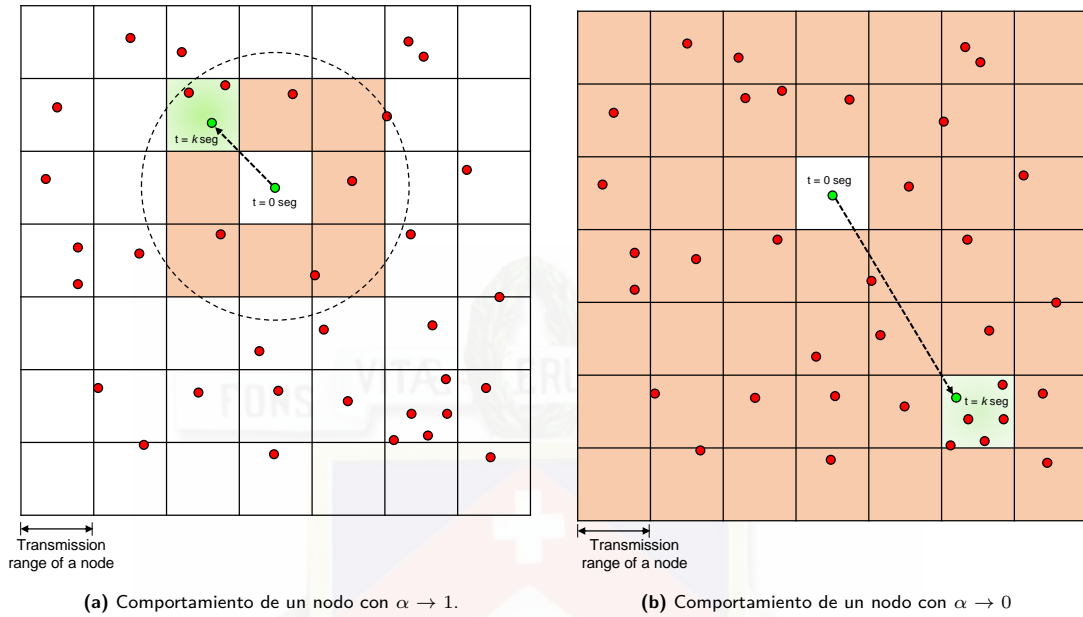


Figura 2.1: Comportamiento modelo **SWIM** en base al valor de α .

este modelo de movilidad utiliza un valor denominado multiplicador de velocidad, una constante que disminuye o incrementa la velocidad del nodo.

En la Fig. 2.1, se muestra el movimiento un nodo. La flecha interpuntada muestra el desplazamiento entre celdas en base al valor de α . Las celdas de color naranja son las tomadas en consideración como destinos y la celda de color verde es la elegida como próximo destino.

El vínculo entre el **ICT** y α , modela el encuentro y tiempo de contacto entre nodos. Es decir, describe un comportamiento de familiaridad, beneficiando la formación de comunidades. Si estos parámetros son sintonizados correctamente los resultados se asemejan mucho a experimentos reales. Gracias a varios trabajos previos [48–50], en [51] se determina que la distribución de tiempo tiene una pendiente de 1,35, el cual es utilizado como el exponente de tiempo de espera. Este valor es tomado en consideración para este trabajo.

SWIM es sencillo de implementar y representa notables rasgos de comportamiento social, por lo que ha sido tomado en cuenta para utilizarlo en este trabajo. Los autores desarrollaron un emulador de código abierto implementado en C++, disponible en [52].

2.3.3. Caminata humana de acción mínima auto-similar (Self-Similar Least-Action Human Walk **SLAW**)

SLAW [9], es un modelo de movilidad que implementa métricas modernas que describen el comportamiento humano. Estas métricas son las siguientes: tiempos de detenimiento, tiempos de inter contacto (**ICT**) y distribución truncada de ley de potencias; las cuales se han evidenciado en varios trabajos sobre movilidad humana [53–56]. En base a estas métricas, **SLAW** pretende representar escenarios sociales como campus universitarios, centros comerciales, restaurantes y parques temáticos.

SLAW presenta en su modelo las siguientes características:

- *Viajes con la ley de potencias truncada y tiempos de pausa*: la distribución truncada de ley de potencias es utilizada para modelar tanto las longitudes de vuelo como en el tiempo de detención en una caminata. Las longitudes de vuelos son los viajes en línea recta sin cambio de dirección, como se explica y modela en [53, 54].
- *Áreas de movilidad acotadas heterogéneamente*: describe el comportamiento de una persona, dependiendo de su personalidad y entorno que lo rodea. Por tanto, tiene distintas posturas de movilización dentro de un región específica [54].
- *Tiempos de inter-contacto de la ley de la potencia truncada*: la distribución truncada de ley de potencias también es utilizada para modelar el intervalo de tiempo en el cual un par de dispositivos establecen contacto [55].
- *Puntos de referencia fractal*: en [56] lo definen como lugares que pueden representar popularidad a los cuales los nodos se sienten atraídos.

Una característica clave de **SLAW** es la capacidad obtener trazas de movilidad tanto de viajes regulares (personas que se desenvuelven en oficinas de trabajo o tienen una rutina cotidiana definida) como de viajes esporádicos (personas que escogen lugares al azar para realizar una visita). Esta característica permite asemejarse más al comportamiento humano y diferenciarse de otros modelos como **RWM** o **TLW**.

La emulación de **SLAW** es implementada y distribuida por los autores en código fuente Matlab, disponible en [57]. Sus principales parámetros de entrada son:

- Valor de *Hurst*
- Número de puntos de referencia
- Radio del grupo
- Rango de agrupamiento
- Peso de la distancia

- Exponente de Levy

El valor de *Hurst* [56] es utilizado para generar puntos de referencia fractales. En [58] se muestran los valores que puede tomar, de tal manera que el comportamiento resultante sea muy semejante a una interacción social. El valor de *Hurst* en conjunto con el parámetro de número de puntos de referencia, sirven para establecer puntos de interés. Estos son distribuidos en toda el área de simulación, cada uno en una cierta área delimitadora establecida por el radio de punto de referencia.

Cabe destacar que cada punto de referencia tiene una probabilidad para ser visitada por un nodo, que se representa con un peso. El punto de referencia con peso más alto tiene mayor posibilidades de ser visitado por un nodo cercano a él. El peso es calculado con: el número de nodos que se encuentren en esa región o hayan pasado por ella, y el valor del peso de la distancia. La finalidad del valor del peso de la distancia es obtener un comportamiento social o esporádico.

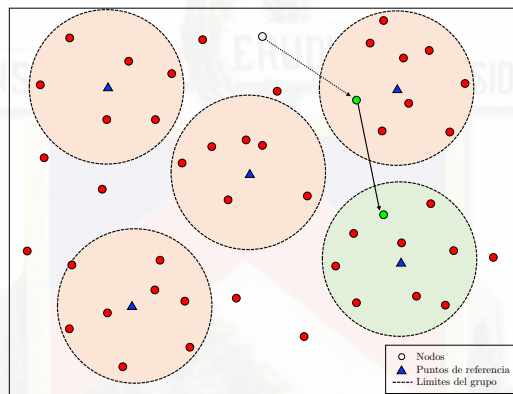
Existe una relacion para la configuracion del peso de la distancia α (especifica decision para que un nodo seleccione su próximo destino) y el exponente de Levy β (especifica el tiempo en el que un nodo permanece detenido) que se muestra en la siguiente ecuación [9]:

$$\alpha + \beta \approx 1 \quad (2.2)$$

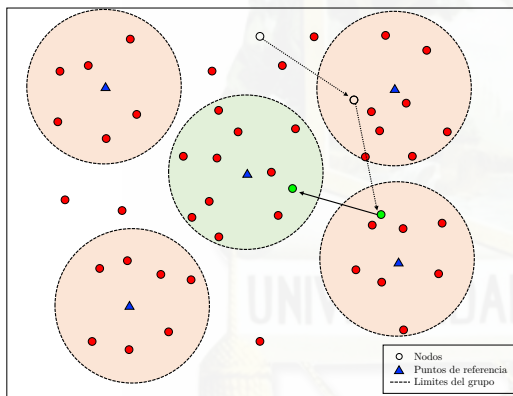
En este modelo, la agrupación de nodos se lleva a cabo de acuerdo al lugar de ubicación de los puntos de referencia. Para esto, es necesario establecer un radio de grupo (cantidad de nodos que se ubicarán alrededor de un punto de referencia) y rango de agrupamiento (distancia en metros del alcance de grupo).

En la Fig. 2.2, se muestran los desplazamientos de un nodo con el modelo [SLAW](#). El nodo se desplaza a través de distintos puntos de referencia. Para cada movimiento, escoge el punto de referencia más cercano y con mayor peso. Cada punto de referencia tiene su respectiva área delimitadora. La región de color verde es la seleccionada como próximo destino.

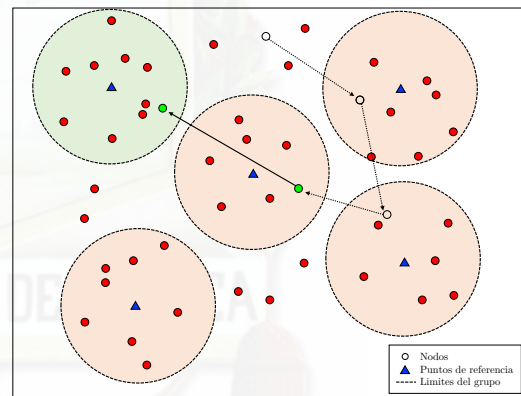
La gran cantidad de parámetros que conforman este modelo lo convierte en una herramienta muy versátil, que puede simular un comportamiento específico deseado. Por ejemplo, [SLAW](#) puede modelar un comportamiento social que desencadene en la formación de comunidades. Por tanto, es tomado en cuenta para la generación de trazas en este trabajo.



(a) Segundo movimiento.



(b) tercer movimiento.



(c) Cuarto movimiento.

Figura 2.2: Movimientos de un nodo a lo largo del tiempo para el modelo de movilidad [SLAW](#).

2.3.4. SMOOTH

SMOOTH [10], es un modelo de movilidad realístico y simple comparado con SLAW. Sin embargo no pierde ninguna de las características de movilidad y comportamiento humano mencionados en [9]. Uno de los principales aportes de este modelo es que no requiere de parámetros complicados de estimar como es el caso del parámetro de *Hurst*. En SMOOTH este parámetro se obtiene utilizando aproximaciones, con el objetivo de alcanzar resultados semejantes a SLAW. Estas aproximaciones se basan en [58], donde se establecen rangos para un tamaño máximo de grupo y distancia máxima del punto de referencia de un grupo a otro [10].

El movimiento de nodos en SMOOTH se basa en patrones de comportamiento social y formación de comunidades. Por ejemplo, un edificio donde existen oficinas que son visitadas con mayor frecuencia que otras (como una sala de conferencias). Por tanto, el modelo forma grupos de distinto tamaño y los distribuye de forma aleatoria y uniforme.

La emulación de este modelo se encuentra implementada en C++, disponible en [59]. Los principales parámetros de entrada son:

- Rango de transmisión de un nodo
- Numero de grupos
- Distribución de viajes
- Distribución de tiempos de detención

Este modelo de movilidad pretende imitar comportamiento social realizando agrupamientos de nodos en distintas regiones. Mediante parámetros de entrada se establece la cantidad de grupos y cada uno con un distinto valor de popularidad. La popularidad define la frecuencia con la que será visitado un grupo. Por tanto, un grupo con mayor popularidad tendrá mayor probabilidad de ser escogido como lugar destino para un nodo. Cabe resaltar que el rango de transmisión de un nodo solo es utilizado por el modelo para lograr asociar un nodo a un grupo.

Cuando un nodo decide su próximo grupo de destino, el modelo calcula la longitud de viaje y la velocidad de desplazamiento, mediante parámetros que describen el comportamiento asintótico de la distribución de viajes [45]. Una vez que el nodo arriva a su grupo, permanece en él por un tiempo. Este tiempo se obtiene utilizando parámetros que describen el comportamiento asintótico de la distribución de tiempos de detención [46].

Este modelo de movilidad implementa características del comportamiento social similares a SLAW. Por tanto, es importante emplearlo en nuestro trabajo.

2.3.5. Área de desastre

Un modelo de movilidad peculiar que no es posible representar utilizando características de comportamiento social es propuesto en [11, 60]. Los autores proponen un modelo de movilidad para un área de desastre. El objetivo es obtener un escenario específico donde se puedan poner a prueba sistemas de comunicación, con la finalidad de verificar su funcionamiento óptimo y eficiente. Este modelo se basa en una táctica de rescate llamada “separación de espacio” (separation of the room). Esta táctica es necesaria para distribuir lugares de atención para las víctimas de un desastre y la ubicación del personal rescatista (bomberos, médicos, etc.).

Las distintas áreas que son consideradas en este modelo son las siguientes: sitio de incidente, área de tratamiento de víctimas, área de compensación de víctimas, zona de transporte (área de ambulancias y helicópteros) y zona de comando técnico operacional [11]. Dentro del área de tratamiento de víctimas existen dos sitios particulares; el primero es el lugar donde las víctimas son rescatadas y esperan ingresar al área de tratamiento, y el segundo es, la estación de salida de damnificados. Finalmente, las personas esperan a ser transportados por ambulancias o helicópteros hacia los hospitales mas cercanos para culminar con su tratamiento.

Las características principales que se debe cumplir el modelo de acuerdo al escenario propuesto, son las siguientes:

- Movimiento heterogéneo dependiente de las áreas: los nodos tienen diferente comportamiento dependiendo del área donde se encuentren.
- Movimiento en caminos óptimos evitando obstáculos: los nodos analizan los caminos posibles y seleccionan el óptimo.
- Los nodos se unen y abandonan áreas específicas: las víctimas y personal rescatista se mueven entre distintas áreas.
- Movilidad grupal (opcional): dependiendo de la ubicación o rol del nodo, éste tiene la opción de moverse grupalmente, en cuyo caso existe un nodo líder del grupo.

Es primordial que un sistema de comunicaciones funcione apropiadamente en escenarios de desastre. Por tanto, se hace factible la evaluación de técnicas de control de topología en estos entornos. Por ende, este modelo es tomado en cuenta para este trabajo.

2.3.6. Restricciones Geográficas

Las limitaciones geográficas no son tomadas en cuenta por modelos de movilidad como: [SWIM](#), [SLAW](#) y [SMOOTH](#). Estos modelos se encuentra libre de escalas y no muestran variaciones,

independientemente de cualquier restricción geográfica. En [61] agregan dicha restricción, obteniendo el modelo denominado “Caminata humana de acción mínima auto-similar basada en mapas” (Map-based Self-Similar Least-Action Human Walk [MSLAW](#)). Con esta característica se agrega un impacto en los resultados de rendimiento para las simulaciones de redes móviles. Adicional, se posibilita también la interacción con los modelos de propagación debido a la distribución de los edificios y la disposición de obstrucciones ambientales.

Para la emulación de este tipo de restricción se requiere de mapas de calles de acceso libre (Open Street Map [OSM](#)) [62]. Además se requiere un emulador que ubique los nodos móviles en ciertas áreas donde sea posible la movilización. En [61] se propone un emulador que en este momento se encuentra fuera de uso (debido a un servidor fue de servicio).

Añadir esta característica dentro de un modelo de movilidad es de gran interés para la evaluación de resultados. La mayor parte de entornos reales, se encuentran delimitados geográficamente. Por lo tanto, el modelo de “Restricciones geográficas” es tomado en cuenta en este trabajo para generar trazas.

2.4. Detección de comunidades

En el contexto de la teoría de grafos, a diferencia de lo que algunas personas piensan, una detección de comunidades no es sinónimo de particionamiento de la red. El particionamiento de red se basa en la división de la red en grupos de aproximadamente igual tamaño. Sin embargo esta no es la idea de la detección de comunidades ya que en la vida real, las comunidades raramente tienen igual tamaño y porque la partición de red no considera similitudes entre usuarios [63]. Por otra parte, en el problema de partición de grafos se conocen de antemano el número de agrupaciones y sus tamaños, mientras que en un problema de detección de comunidades no se conocen dichos valores por lo que la detección permite *descubrir* las propiedades de la red.

Existen muchas definiciones de lo que es una comunidad dentro de un grafo. Fortunato en [64] distingue tres tipos de definiciones de comunidades: definiciones locales, definiciones globales y definiciones basadas en similaridad de vértices.

Las definiciones locales se enfocan en el subgrafo en cuestión, incluyendo posiblemente a sus vecinos inmediatos, pero despreciando al resto del grafo. Bajo este contexto, las comunidades se consideran entidades separadas con su propia autonomía. Por ejemplo, en [65] se presenta un método centralizado de detección de comunidades locales. Primero se forman comunidades tempranas con nodos de alto grado. Estas comunidades tempranas forman los núcleos principales de una comunidad real, por lo que posteriormente, se fusionan en comunidades más grandes. Se utiliza una medida para determinar si una comunidad es independiente de otra comunidad

vecina o si es necesario una fusión. Finalmente el algoritmo retira los nodos que se encuentren en más de una comunidad (comunidades sobrepuestas) en caso que así se lo requiera. La complejidad computacional de este método es de $O(kn^2)$ siendo k el grado máximo de los nodos del grafo, y n el número de nodos.

Las definiciones globales en cambio, toman en cuenta al grafo como un todo. Hay una clase de definiciones apropiadas basadas en la idea que un grafo tiene estructura en comunidades si es que es distinto a un grafo aleatorio. Bajo esta idea también, Newman y Girvan [19] definen la modularidad, la cual es la función más popular hoy en día para medir la calidad de particiones. Se hace una comparación entre la densidad de aristas en un subgrafo, y la densidad que se esperaría tener en el mismo subgrafo si los vértices del grafo fueran enlazados independientemente de la estructura en comunidad. En otras palabras, la modularidad mide la diferencia entre la densidad de aristas dentro de la comunidad y la densidad que se obtendría si las aristas fueran distribuidas aleatoriamente en la red. Para redes ponderadas, la modularidad Q es calculada con la siguiente fórmula [19]:

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j) \quad (2.3)$$

Donde A es la matriz de adyacencia, A_{ij} representa el peso de enlace entre i y j , $k_i = \sum_j A_{ij}$ es la suma de los pesos de las aristas conectadas al nodo i , c_i es la comunidad a la cual i es asignada, la función $\delta(u, v)$ es 1 si $u = v$ y 0 caso contrario, y $m = \frac{1}{2} \sum_{i,j} A_{ij}$ es el número total de aristas en el grafo.

Finalmente las definiciones basadas en la similaridad de los vértices dicen que cada vértice pertenece al cluster en el cual los vértices son lo más similares a él. Un ejemplo de medida de similaridad puede ser la distancia entre un par de vértices.

Entonces, dado un grafo, el problema de detección de comunidades es el de asignar a los nodos a comunidades dentro de las cuales sus enlaces sean altamente relacionados y cuyos enlaces hacia otras comunidades sean débiles. Numerosos algoritmos han sido propuestos a lo largo del tiempo para hallar estas particiones. Los algoritmos de detección de comunidades óptimos, que devuelven los mejores resultados posibles, son NP-hard. Por ejemplo aquellos basados en maximización de modularidad, por lo que se prefiere utilizar algoritmos de aproximación [64]. Gran parte de estos algoritmos se centran en redes binarias, es decir, en redes cuyas aristas tienen peso 1 en caso de existir y 0 caso contrario [66]. Este tipo de redes se manifiestan en algunos tipos de entornos como por ejemplo en redes biológicas en donde el enlace entre dos nodos o bien existe o no. Sin embargo, la mayoría de las redes en la vida real son ponderadas y por tanto una detección de comunidades acertada debe ser capaz de tomar en cuenta dichos

pesos.

2.5. Métricas de centralidad

Uno de los principales asuntos del análisis de redes sociales es la de identificar a los nodos más importantes dentro de la red, utilizando métricas de teoría de grafos. Las métricas de centralidad ayudan a identificar los actores más relevantes en una red [67]. Para el presente estudio, las métricas de centralidad más relevantes pertenecen a dos categorías. **Las métricas basadas en el grado del nodo** consideran a un nodo prominente si los vínculos del mismo lo hacen visible para los demás actores de la red. Las principales métricas basadas en el grado del nodo son *Centralidad de grado* y *Centralidad Espectral*. Por otro lado, **Las métricas basadas en las distancias geodésicas de los actores** tienen muchas variantes debido a que los caminos más cortos son frecuentemente utilizados en tareas de red. Las más populares pertenecientes a ésta categoría son *Centralidad de Cercanía* y *Centralidad Betweenness*.

Centralidad de Grado- Se define como el número de vecinos (1 salto) de un actor. La centralidad de grado C_D correspondiente al nodo v se calcula de la siguiente manera [68]:

$$C_D(v) = \frac{d(v)}{n-1} \quad (2.4)$$

Donde $d(v)$ es el grado del nodo v y n es el número de nodos en la red.

Centralidad Espectral- Toma en cuenta el número y la calidad de los nodos vecinos [67]. Esta métrica define la prominencia de un actor de manera recursiva. La más popular es la métrica *Page-Rank* [69]:

$$C_{PR}(v_i) = \alpha \frac{1}{n} + (1 + \alpha) \sum_{v_j: v_j \rightarrow v_i} \frac{C_{PR}(v_j)}{k_{out}(v_j)} \quad (2.5)$$

Donde α es un valor escalar entre (0,1), y $k_{out}(v_j)$ es el número de aristas salientes del nodo v_j .

Centralidad de Cercanía- Mide el tiempo de propagación de información desde un nodo hacia los demás utilizando la distancia geodésica. Para el nodo v_i en una red de n nodos, la centralidad de cercanía C_C se calcula de la siguiente manera [68]:

$$C_C(v_i) = \frac{n-1}{\sum_{j \neq i} distance(v_i, v_j)} \quad (2.6)$$

Centralidad Betweenness- Se define como la fracción de caminos más cortos entre un par

de nodos cualesquiera de una red que pasan a través de un nodo. Representa la medida en la que ese nodo tiene control sobre la información que fluye entre otros nodos [68]. La centralidad betweenness C_B de un nodo v es:

$$C_B(v) = \sum_{j=1}^N \sum_{k=1}^{j-1} \frac{g_{jk}(v)}{g_{jk}} \quad (2.7)$$

Donde, g_{jk} es el número de caminos más cortos entre j y k , $g_{jk}(v)$ es el número de caminos más cortos entre j y k que pasan por v y N es el número total de nodos.

En [66] se definen dos métricas de centralidad derivadas de la centralidad betweenness: Intra-centralidad e Inter-centralidad, que se detallan a continuación.

La **Intra-centralidad** $intraC_B$ de un nodo v mide la influencia del nodo dentro de su comunidad. Se obtiene calculando su centralidad betweenness, pero considerando únicamente los nodos que pertenezcan a la misma comunidad de v , es decir:

$$intraC_B(v) = \sum_{j=1}^{N(C_v)} \sum_{k=1}^{j-1} \frac{g_{jk}(v)}{g_{jk}}, \quad i, j \in C_v \quad (2.8)$$

donde C_v es la comunidad a la que pertenece v , y $N(C_v)$ es el número total de nodos que pertenecen a C_v . Si un nodo pertenece a más de una comunidad (en el caso de comunidades superpuestas) entonces tiene un valor de $intraC_B$ por cada comunidad a la que pertenezca.

La **Inter-centralidad** $interC_B$ de un nodo v para dos comunidades C_m y C_n se define como la centralidad betweenness de ese nodo considerando únicamente los nodos que pertenezcan a cualquiera de esas comunidades. En general este valor mide la capacidad de un nodo para comunicar a las dos comunidades en cuestión:

$$interC_B(v) = \sum_{j=1}^{N(C_m \cup C_n)} \sum_{k=1}^{j-1} \frac{g_{jk}(v)}{g_{jk}}, \quad i, j \in C_m \cup C_n \quad (2.9)$$

Cada nodo tiene un valor de $interBC(v)$ por cada par de comunidades detectadas en la red.

$intraC_B(v)$ es una buena opción para detectar los nodos más influyentes dentro de una región, sin embargo, la inter-centralidad $interC_B(v)$ parece no ser una opción viable para detectar nodos que comuniquen a dos o más regiones densas. Esto debido a que se requiere del conocimiento

de todas las comunidades en la red y cada nodo debe generar un valor de $interC_B(v)$ por cada par de comunidades detectadas.

En [70] se propone otra derivación de métrica de centralidad betweenness con un concepto similar a inter-centralidad. Se trata de **Centralidad de Puenteado** la cual identifica a los nodos por los cuales fluye mayor cantidad de información y que se encuentran ubicados en medio de regiones densamente conectadas. Es decir, regiones con alta modularidad. La centralidad de puenteado $C_{Brid}(v)$ se obtiene multiplicando la centralidad betweenness $C_B(v)$ y el coeficiente de puenteado $BC(v)$ de un nodo:

$$C_{Brid}(v) = BC(v) \times C_B(v) \quad (2.10)$$

donde $C_B(v)$ es la centralidad betweenness [Equ. 2.7] que mide las características globales del nodo. Mientras que $BC(v)$ es el coeficiente de bridging y mide las características locales del nodo:

$$BC(v) = \frac{d(v)^{-1}}{\sum_{i \in N(v)} \frac{1}{d(i)}} \quad (2.11)$$

donde $d(v)$ es el grado del nodo v y $N(v)$ es el conjunto de vecinos de v . De acuerdo a [70] los nodos que en realidad se consideran “nodos puente” son aquellos que están dentro del 25 % con mayor valor de C_{Brid} en la red.





Capítulo 3

Trabajos relacionados

En este capítulo resumiremos algunos trabajos relacionados con el control de topología, las métricas de centralidad y detección de comunidades, con fin de dirigir nuestro trabajo hacia una combinación de todos estos.

En la actualidad, los mecanismos de control de topología en redes WMNs han generado importancia e interés en la comunidad científica. Esto debido a la amplia rama de investigación provocada, dando lugar a varios trabajos que recopilan técnicas de control de topología. Aziz et al. en [5], revisan varias técnicas de control de topología distribuida, enfocándose principalmente en extender el tiempo de vida de las baterías en las WSNs. Las técnicas se clasifican principalmente en algoritmos de control de topología 2D y 3D. Los algoritmos 2D se clasifican en: ajuste de potencia, modo de operación, agrupamiento, e híbridos. Mientras tanto, los algoritmos 3D surgen debido a la necesidad de implementar control de topología en entornos submarinos o edificios. Por su parte, Huang et al., en [71] presentan una taxonomía de algoritmos de control de topología que logran eficiencia energética, enfocados en la implementación de Internet de las cosas (Internet of Things IoT). Clasifican los algoritmos en tres clases: Homogéneos, Heterogéneos y Otros. Los algoritmos homogéneos y heterogéneos asumen que los dispositivos que conforman la topología son homogéneos y heterogéneos respectivamente. Los otros algoritmos se basan en la localización, dirección, y vecinos de los dispositivos.

La mayor parte de trabajos de control de topología se enfocan en el ajuste de potencia de transmisión de los nodos [35, 37–39, 72]. Por ejemplo, en [38] se propone un algoritmo distribuido en el que cada nodo escoge su potencia de transmisión, garantizando conectividad global de la red. Se basa en información direccional de transmisión. Entonces, un nodo aumenta su potencia hasta tener un vecino en cada una de sus direcciones. Por otro lado, Ma et al. en [72] proponen el algoritmo FTMRT para el control de topología tolerante a fallos para redes WSNs heterogéneas (HWSN). Utilizan árboles de enrutamiento múltiple, con el fin de mantener múltiples rutas y facilitar la reconstrucción de la topología en el caso de que alguna estación falle. El ajuste de potencia de los nodos sensores se lleva a cabo en base al árbol de enrutamiento múltiple.

Algunos trabajos se enfocan en la colocación estratégica de elementos de red para mejorar la topología. Por ejemplo, en [73] hacen una revisión de estrategias de ubicación, capacidad de *buffering*, y modelos de movilidad, para la implementación de *Throwboxes* en redes tolerantes a retardo DTN. *Throwboxes* son dispositivos que actúan como relé o intermediarios entre nodos de una red. La implementación de estos dispositivos tiene como finalidad incrementar la tasa de entrega, reducir el retardo, y disminuir la carga en los nodos móviles de una red.

En el contexto de comunidades, existen varios trabajos que toman en cuenta la estructura comunitaria para mejorar el desempeño de las redes. Por ejemplo, un enfoque es propuesto por Lin et al., en [74] para WMNs con estructuras comunitarias. Presentan un método basado en la ubicación dinámica de enrutadores de tipo malla bajo la suposición de que los nodos conocen la estructura de comunidades de la red. Para determinar la ubicación de los enrutadores, proponen un enfoque de optimización de enjambre de partículas (*Particle Swarm Optimization PSO*) basado en comportamiento social. Como resultado, la característica de agrupamiento mejora la ubicación de enrutadores y por tanto la cobertura y conectividad en la red.

Varios trabajos aplican análisis de redes sociales (SNA) para crear topologías o diseñar algoritmos de enrutamiento. Utilizan métricas de centralidad como: centralidad *betweenness*, centralidad de grado, centralidad de cercanía, entre otras [68]. El trabajo realizado por Kas et al., en [75] muestra cómo los nodos con mayor centralidad tienen un impacto directo en la confiabilidad de redes WMNs. A demás los autores muestran un aumento del rendimiento de una red, al utilizar una asignación prioritaria de ranuras de tiempo (en TDMA), en los nodos con mayor valor de centralidad. En [76] hacen uso de métricas de centralidad para determinar CDSs en grafos complejos grandes. Como resultado, la centralidad *betweenness* logra formar CDSs con menor tamaño en la mayoría de los casos. Por su parte, en [77] se realiza una comparación entre las métricas de centralidad de grado, cercanía y *betweenness* para formar el backbone de una red en malla de 100 nodos distribuidos aleatoriamente. Se escogen el 40 % y el 50 % de los nodos mas centrales para que cumplan las funciones de enrutadores mesh, mientras que el resto de los nodos son únicamente hosts. Para realizar la comparación, se calcula la fragmentación de la red resultante para cada uno de los 3 casos. Como resultado de dicha comparación, la métrica de centralidad *betweenness* consigue menor valor y un comportamiento más estable de fragmentación que las otras métricas, concluyendo que es la mejor opción para redes con distribución aleatoria de nodos.

Existen también trabajos que combinan métricas de centralidad y estructuras de comunidades para mejorar el desempeño de las redes. Bubble Rap [78] es un algoritmo utilizado para el reenvío de paquetes, basado en el comportamiento social en combinacion del conocimiento de la estructura de comunidades y la centralidad de los nodos para realizar decisiones de reenvío. La idea es reenviar los paquetes hacia nodos más populares. Cada nodo tiene un ranking global (dentro de toda la red) y un ranking local (dentro de su comunidad). El emisor envía el mensaje hacia arriba en el árbol jerárquico hasta alcanzar algún nodo que se encuentre en la misma comunidad que el nodo destino. Luego se utiliza el ranking local y el mensaje se reenvía hasta alcanzar al destino o hasta expirarse. La detección de comunidades la realizan con el método distribuido propuesto por Hui et al.[79], y para el cálculo de las centralidades de una manera distribuida, utilizan un método de aproximación. Un algoritmo para el reenvío de paquetes en redes tolerantes a retardo (DTN) se propone en [66], el cual combina el descubrimiento de comunidades formadas y las métricas de centralidad inter-centralidad e intra-centralidad.

La mayor parte de trabajos de control de topología existentes, se enfocan en el ajuste de potencias de transmisión y ubicación geográfica de elementos de red que puedan mejorar el desempeño de la misma. Las métricas de centralidad y la estructura en comunidades se han utilizado independientemente o de manera conjunta para desarrollar algoritmos de control de topología o reenvío de mensajes. En nuestro trabajo proponemos un control de topología basado en métricas de centralidad y estructuras comunitarias formadas debido al comportamiento social entre nodos. Se basa en la selección de un set dominante conectado CDS en una red homogénea. Los



nodos de que no pertenezcan a este [CDS](#), apagarán sus funcionalidades de enrutamiento. Como resultado obtendremos topologías reducidas que serán evaluadas y comparadas con un método de control de topología que únicamente toma en cuenta métricas de centralidad.





Capítulo 4

Control de topología basado en métricas de centralidad y estructura comunitaria

En este capítulo presentamos el diseño de un método de control de topología que se basa en métricas de centralidad y toma en cuenta la estructura en comunidades de la red.

4.1. Escenarios bajo consideración

Para este trabajo se considera utilizar escenarios que representen un entorno de comportamiento social lo más realista posible. Dentro de cada escenario se considera una red inalámbrica en malla con 100 nodos basados en el estándar 802.11s. Los nodos se distribuyen en un área de 1040×520 metros y la duración de las simulaciones es de 500 segundos. Cada escenario tiene la misma base estructural con el fin de disponer de una evaluación objetiva. Los escenarios se diferencian principalmente por el modelo de movilidad que es aplicado en el movimiento de sus nodos. Los seis modelos de movilidad tomados en cuenta son los siguientes:

- [SWIM](#)
- [SLAW](#)
- SMOOTH
- Área de desastre
- Restricciones Geográficas
- Movimientos aleatorios, [RWM](#).

Mismos que han sido explicados en la sección 2.3. Todos los modelos de movilidad a excepción de “restricciones geográficas” son emulados mediante el software Bonnmotion, explicado en la sección 5.1. Para restricciones geográficas se utiliza el software [SUMO](#). A continuación se detallan los parámetros de configuración y las topologías de red resultantes.

4.1.1. Escenario [SWIM](#)

Con [SWIM](#) se pretende generar escenarios que representen un comportamiento de familiaridad e interacción social de los nodos con los vecinos mas cercanos, siguiendo el comportamiento descrito en la sección 2.3.2. Para esto se selecciona un valor de α de 0.85 y se toma en cuenta una distribución de Pareto mostrada en [9], para establecer los limites de tiempo de detención con un mínimo valor de 30 segundo y máximo de 500 segundos. De igual forma, el valor del exponente de tiempo de espera es de 1.35 (tomado de [51]). Los demás parámetros de entrada están basados en experimentos mostrados en [52]. En la Tabla 4.1 se muestra a detalle los parámetros de entrada con sus respectivos nombres de argumento para la simulación del modelo [SWIM](#) en Bonnmotion.

Parámetro	Argumento	Valor
Rango de transmisión del nodo:	-r	0.04
Peso de distancia de la celda:	-c	0.85
Multiplicador de la velocidad del nodo:	-m	0.01
Exponente de tiempo de espera :	-e	1.35
Límite superior de tiempo de espera:	-u	500

Tabla 4.1: Parámetros de entrada para [SWIM](#).

4.1.2. Escenario [SLAW](#)

Con [SLAW](#) se pretende emular un centro comercial, en el cual los nodos móviles (transportados por personas) se puedan desplazar libremente a través de varios puntos de interés. Para este caso, los puntos de interés representan tiendas del centro comercial. Para establecer el valor de los parámetros de entrada nos basamos en simulaciones realizadas en [9], y en pruebas realizadas en [58] para la sintonización del parámetro de *Hurst*. Los parámetros de entrada y sus nombres de argumentos respectivos para el ingreso de estos valores en Bonnmotion se detallan en la Tabla 4.2.

Parámetro	Argumento	Valor
Numero de puntos de referencia:	-w	50
Tiempo mínimo de pausa:	-p	30
Tiempo máximo de pausa:	-P	500
Exponente Levy:	-b	1
Parámetro de Hurst :	-h	0.75
Peso de la distancia:	-l	3
Rango de agrupamiento:	-r	50
Radio del grupo:	-Q	10
Radio de los puntos de referencia:	-W	10

Tabla 4.2: Parámetros de entrada para [SLAW](#).

4.1.3. Escenario SMOOTH

Mediante SMOOTH se pretende emular un escenario similar a [SLAW](#) que represente un centro comercial con varias tiendas manifestadas como puntos de referencia. Para la configuración de parámetros de entrada nos basamos en simulaciones efectuadas en [10]. Por tanto, los parámetros

de entrada utilizados en este trabajo para este modelo están detallados en la Tabla 4.3.

Parámetro	Argumento	Valor
Rango del nodo:	-g	100
Numero de clusters:	-h	20
Comportamiento asintótico de la distribución de viajes (α, f_{min}, f_{max}):	-k, -l, -m	0.7, 1, 500
Comportamiento asintótico de la distribución de tiempos de detención(α, f_{min}, f_{max}):	-o, -p, -q	2, 1, 1000

Tabla 4.3: Parámetros de entrada para SMOOTH.

4.1.4. Escenario de área de desastre

Se basa en una catástrofe real, que tuvo lugar en mayo de 2005 en el país de Alemania, durante la preparación de la Jornada Mundial de la Juventud 2005 y la Copa Mundial de Fútbol de la FIFA en 2006, dentro de un salón de eventos [11]. A continuación se describe la configuración para la emulación de este escenario.

Se asume nodos con velocidad peatonal ente 1 y 2 m/s. El tiempo de detención de cada nodo está entre 0 y 20 segundos. La probabilidad de cambio de grupo se establece en 0 debido a que ningún nodo víctima se convertirá en rescatista o paramédico. Parámetros como la media de nodos por grupo, máxima distancia al centro del grupo y máximo tiempo de detención se modelan de acuerdo al escenario establecido en [11]. Para la simulación se utiliza BonnMotion (tomado de los ejemplos proporcionado por el software) con los parámetros detallados en la Tabla 4.4.

Para generar el movimiento heterogéneo, el área de simulación se divide en sub-áreas siguiendo la clasificación de tácticas de rescate descrito en la Sección 2.3.5. Estas áreas se clasifican en:

- Ubicación del incidente (IL)
- Pacientes en espera del área de tratamiento (PWT),
- Estación de compensación de víctimas (CCS)
- Punto de estacionamiento de ambulancias
- Comando técnico operacional (TOC).

Cada una de estas áreas debe tener un punto de entrada y salida (e_r y a_r). Además cada área tiene un número de nodos y ciclo de movimiento acorde a los nodos existentes dentro del área.

Los polígonos y demás parámetros que forman estas áreas se detallan en la Tabla 4.5. En la Fig. 4.1 se observa el movimiento de 100 nodos durante 500 segundos. Se puede distinguir las regiones y recorridos que toman los nodos por los distintos puntos de entrada y salida de las regiones siguiendo el camino más corto.

Parámetro	Argumento	Valor
Media de nodos por grupo:	-a	1
Probabilidad de cambio de grupo:	-c	0
Áreas de catástrofe:	-e	8
Máxima distancia al centro del grupo:	-r	1
Máxima velocidad:	-h	2
Mínima velocidad:	-l	1
Máximo Tiempo de detención:	-p	20

Tabla 4.4: Parámetros de entrada para Área de Desastre.

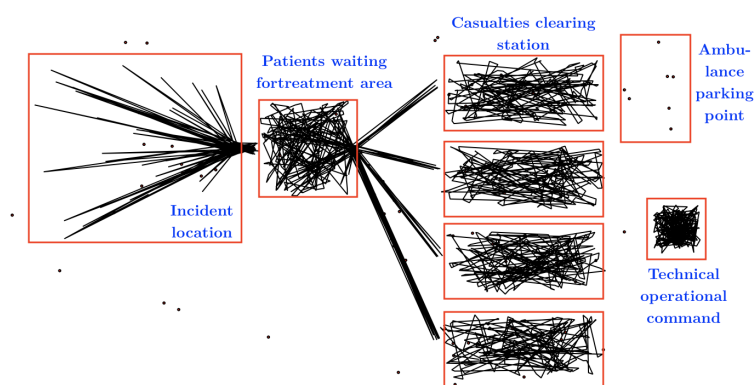
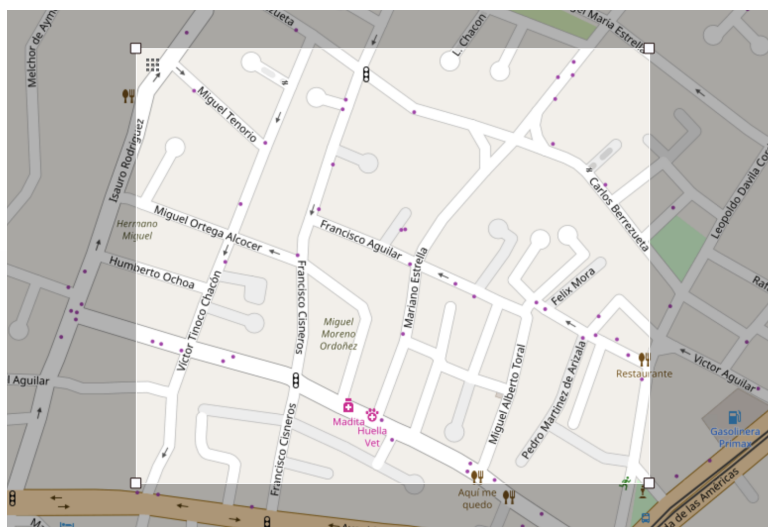


Figura 4.1: Traza resultante del modelo de área de desastre.



Parámetro	IL	PWT	CCS ₁	CCS ₂	CCS ₃	CCS ₄	TOC	APP	OBST
x_1 :	25	133	220	200	220	220	320	320	25
y_1 :	5	25	5	46	86	126	75	5	125
x_2 :	125	180	300	300	300	300	345	345	100
y_2 :	5	25	5	46	86	126	75	5	125
x_3 :	125	180	300	300	300	300	100	50	100
y_3 :	100	75	40	80	120	160	320	50	200
x_4 :	25	133	220	220	220	220	345	320	25
y_4 :	100	75	40	80	120	160	100	50	200
e_{rx} :	25	133	220	220	220	220	330	5, 345	-
e_{ry} :	50	50	20	60	100	140	75	0, 0	-
a_{rx} :	125	180	310	310	310	310	330	325, 325	-
a_{ry} :	50	50	20	60	100	140	76	5, 50	-
Número de nodos:	15	39	15	15	15	15	6	30	-
Ciclo de movimientos:	15	37	-	-	-	-	-	28	-

Tabla 4.5: Disposición de sub áreas estratégicas para el área de desastre.



(a) Extracto de mapa obtenido de OSM.



(b) Región permitida para la circulación de nodos en SUMO.



(c) Trayectorias para 100 nodos en ns-2.

Figura 4.2: Proceso de obtención de la traza con el escenario de restricciones geográficas.

4.1.5. Escenario de restricciones geográficas

Con este modelo se pretende emular un escenario que restringe los caminos para la circulación de un nodo, utilizando un mapa real. En la Fig. 4.2 se observa los planos utilizados para la obtención del modelo de restricciones geográficas. La región extraída de OSM [62] pertenece a una zona de la ciudad de Cuenca en Ecuador, como se observa en la Fig. 4.2a. Mediante el software SUMO se obtienen las regiones permitidas para la circulación de nodos peatones, señaladas de color azul en la Fig. 4.2b. Adicionalmente, con este software se generan los viajes, utilizando una tasa de aparición de 0.2 nodos por segundo durante 500 segundos de simulación y la distancia máxima recorrida por un nodo es de 800 metros. Finalmente, en la Fig. 4.2c se muestran las trayectorias resultantes para 100 nodos.

4.1.6. Escenario RWM

Finalmente, el modelo de punto de referencia aleatorio [RWM](#) ha sido utilizado frecuentemente para el análisis de redes ad hoc. En este modelo, cada nodo elige al azar un destino y se mueve en línea recta con una velocidad escogida al azar de un intervalo entre v_{min} y v_{max} . Cuando el nodo alcanza su destino, permanece estacionario durante un tiempo de pausa predefinido y luego comienza nuevamente a moverse aleatoriamente. Para este trabajo se selecciona la velocidad de movimiento entre 2 y 4 m/s y un tiempo de detención de 20 segundos.

4.2. Detección de comunidades

Como ya hemos mencionado, una vez que los nodos de la red adquieren patrones de movilidad humana, éstos forman comunidades debido al comportamiento social de las personas. Los nodos mantienen una mejor interacción temporal y espacial con aquellos nodos pertenecientes a su misma comunidad. Por tanto, el primer paso para diseñar un mecanismo de control de topología consiente de estructura comunitaria, es detectar éstas comunidades formadas en cada escenario a lo largo del tiempo. A continuación detallamos el algoritmo que se ha utilizado para la detección de comunidades en este trabajo.

En [80] se presenta el algoritmo de Louvain para la detección de comunidades. Es un algoritmo centralizado capaz de detectar comunidades en redes ponderadas de manera global y se basa en la maximización de la modularidad. El algoritmo se divide en dos fases que se repiten de manera iterativa. En la primera, se asigna una comunidad distinta a cada nodo en la red. Luego, para cada nodo i se consideran sus vecinos j y se evalúa la ganancia de modularidad que se lograría si i se moviese a la comunidad de j . Entonces se procede a mover a i a la comunidad que logre la máxima ganancia de modularidad. Si no es posible ninguna ganancia positiva, i se queda en la comunidad en la que se encuentra. Esto se realiza de manera repetida y secuencial para todos los nodos hasta que no se puedan lograr mejoras, completando así la primera fase del algoritmo.

La segunda fase consiste en construir una nueva red cuyos nodos son las comunidades que se formaron en la primera fase. Para ello, los pesos de los enlaces entre los nuevos nodos son dados por la suma de los pesos de enlaces entre nodos de las dos comunidades. Una vez completada la segunda fase, es posible aplicar la primera fase a la red resultante. Las fases se repiten de manera iterativa hasta que no se logren nuevos cambios y se haya conseguido una modularidad máxima. Este método resulta ser muy intuitivo y simple, lo que lo convierte en un algoritmo muy rápido. Su limitante principal es el espacio de almacenamiento de las redes en la memoria (para redes muy grandes), mas no el tiempo de computación. Su complejidad computacional es

de orden $O(M)$ por iteración.

En [81] se presentan heurísticas paralelas para la detección de comunidades de una manera aún más rápida utilizando el algoritmo de Louvain [80] como plantilla. Este método logra obtener comunidades con mayor modularidad y en tiempos hasta 16 veces (utilizando 32 hilos) más rápidos. Esto como resultado de analizar los nodos de manera paralela. Una heurística que se utiliza para combatir los desafíos que se presentan en el enfoque paralelo, es la implementación de un esquema de etiquetado en el que se asignan etiquetas numéricas a las comunidades en un orden arbitrario en cada fase del algoritmo. Este método es utilizado a lo largo de este trabajo para detectar las comunidades en los grafos, debido a su bajo costo computacional y buenos resultados obtenidos. El algoritmo se encuentra desarrollado en código C++ y OpenMP bajo el nombre de *Grappolo* y su descarga está disponible bajo licencia BSD modificada de 3 cláusulas [82].

Los pesos de las aristas son dados por $\max(c_a) - c_a$, donde c_a es la métrica Enlace Tiempo-en-Aire definida en la Ec.(2.1) y $\max(c_a)$ es el máximo valor de c_a en toda la red. Entonces, $\max(c_a) - c_a$ define qué tan fuerte es un enlace entre dos nodos. A manera de ilustración, en la Fig. 4.3 se pueden ver las comunidades detectadas en una captura instantánea de una simulación utilizando el modelo de movilidad SWIM. Las comunidades están distinguidas varios colores. Para evaluar la calidad de detección de comunidades, en la Tabla 4.6 se observa la modularidad obtenida a través del tiempo de los distintos modelos de movilidad. Para un mayor detalle, en el Apéndice A se presentan las gráficas evolutivas del número de comunidades detectadas y modularidades alcanzadas para todos los escenarios.

Modelo	min	max	media	std
SWIM	0.5952	0.9465	0.769	0.1069
SLAW	0.5456	0.9345	0.7081	0.1219
SMOOTH	0.6183	0.9258	0.7307	0.0945
Disaster A.	0.651	0.9488	0.7437	0.0972
Geographic R	0.5548	0.9283	0.7217	0.0992
RWM	0.6278	0.9115	0.7335	0.089

Tabla 4.6: Modularidades alcanzadas con detección de comunidades

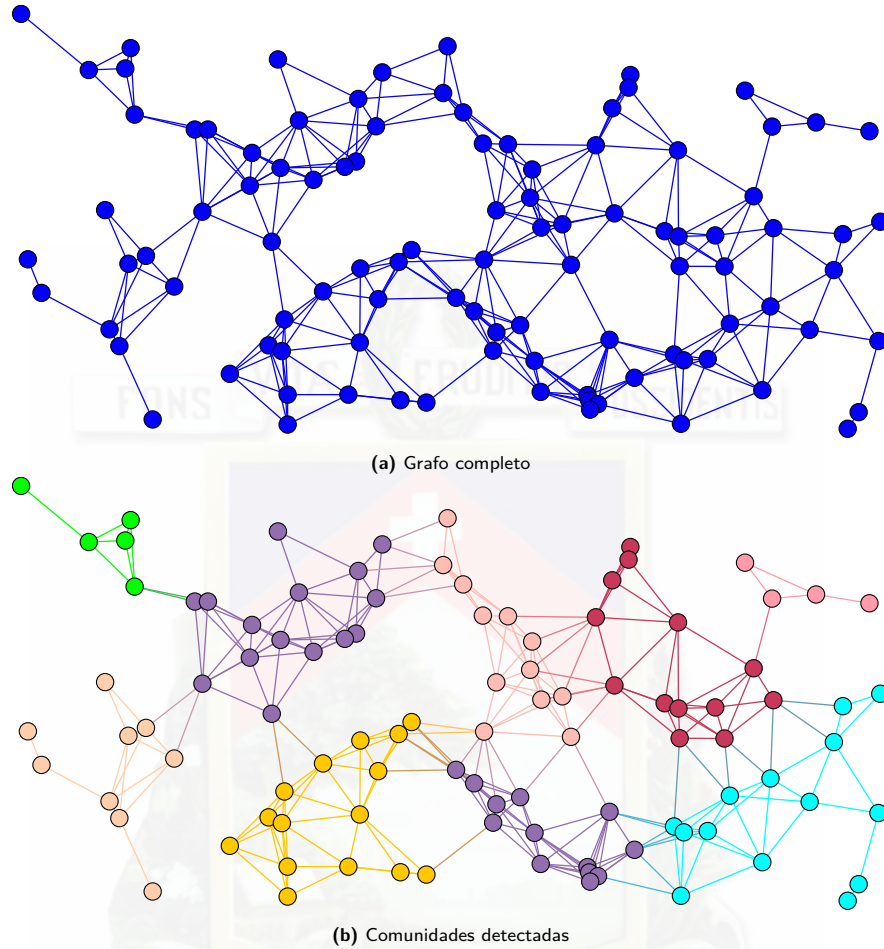


Figura 4.3: Topología de la red para el escenario con modelo de movilidad SWIM en $t = 30\text{seg}$

4.3. Selección de Enrutadores

En la Sección 2.5 se analizó y concluyó que las métricas de centralidad *Betweenness* y sus variantes son las preferidas para el control de flujo y reenvío de información a través de una red. Básicamente mientras mayor valor de C_B tenga un nodo, mayor es el control que éste tiene sobre la información que fluye entre los nodos. Tomando en cuenta que la topología sujeta a estudio tiene una estructura en comunidades debido a los distintos modelos de movilidad aplicados, se vuelve evidente que el diseño de control de topología debe tomar en cuenta a dichas comunidades. En base a estudios previos [70, 77], Consideramos las métricas de centralidad *Betweenness* y de puentado para la selección de enrutadores malla. Específicamente, en este trabajo se propone escoger a los nodos más centrales dentro de cada comunidad, así como a los nodos que de mejor manera comunican a éstas comunidades entre sí, para que cumplan las funciones de enrutadores. En este contexto, se proponen y evalúan tres distintos métodos de

selección de enrutadores: k nodos más centrales en cada comunidad, C -A HI nterA C_B N y C -A HC_B N que serán explicados en la siguiente sección.

Al igual que en un trabajo previo [77], para evaluar en primera instancia la calidad de cada uno de los métodos, procedemos a estimar dos características de las topologías resultantes. En primer lugar se evalúa la fragmentación del backbone resultante, es decir el grafo que se genera al considerar solamente los nodos escogidos como enrutadores. La fragmentación de un grafo se define como la porción de nodos que se encuentran desconectados el uno del otro y se calcula mediante la siguiente ecuación [83]:

$$F = 1 - \frac{\sum_k s_k(s_k - 1)}{N(N - 1)} \quad (4.1)$$

Donde s_k es el tamaño del k -ésimo componente del grafo y N es el número total de nodos del grafo. Se supone que el backbone producido debe tener una fragmentación cero, es decir, que todos los nodos seleccionados como enrutadores deben tener conexión ya sea directa o indirecta entre ellos. En caso contrario habrá un backbone fragmentado y sin conexión entre todos sus elementos. Cabe recalcar que el hecho que el backbone no esté fragmentado, no significa necesariamente que no existan nodos aislados. Por esta razón, en segundo lugar se procede a evaluar el número de nodos aislados para cada método propuesto. Estas dos medidas permiten tener una idea del mejor método en cuanto a conectividad de red. Para la evaluación de las topologías resultantes con cada uno de los métodos propuestos, mostraremos los resultados correspondientes a una captura instantánea ($t = 30\text{seg}$) del escenario con modelo de movilidad $SWIM$.

4.3.1. k nodos más centrales en cada comunidad

Este método consiste en elegir en cada comunidad el o los k nodos con mayor valor de centralidad Betweenness C_B , y un cierto porcentaje de nodos con mayor valor de centralidad de puentado C_{Brid} de toda la red. Estos nodos elegidos serán los que cumplan la función de enrutadores de malla (MR). En la Fig. 4.4a se puede observar la fragmentación del backbone resultante para el caso de escoger uno ($k = 1$, con color azul) o los dos nodos ($k = 2$, con color rojo) con mayor C_B en cada comunidad conjuntamente con un porcentaje (entre 20 % y 80 %) de nodos con mayor valor de C_{Brid} . Para este caso específico, la fragmentación del backbone alcanza un valor de cero estable cuando se escogen el 39 % o más de los nodos con mayor C_{Brid} en la red ya sea para $k = 1$ o $k = 2$. Para la red sujeto de estudio con 100 nodos, significa que al escoger los 39 nodos con mayor C_{Brid} de toda la red y el nodo con más C_B de cada comunidad, se alcanza una conectividad total en el backbone formado.

Ahora bien, podrían existir nodos aislados que se encuentren en los extremos de la red. Como

estos nodos posiblemente no pertenezcan al backbone, no afectan a la fragmentación analizada anteriormente. En la Fig. 4.4b por su parte, se observa el comportamiento de éste parámetro. Se puede observar que para que no existan nodos aislados en la red, se deben escoger el 63 % de nodos con mayor valor C_{Brid} . En una red de 100 nodos, 63 nodos que funcionen como enrutadores no presentaría un impacto real de control de topología y por tanto se puede concluir que *k nodos más centrales en cada comunidad* no es un buen método de selección de enrutadores.

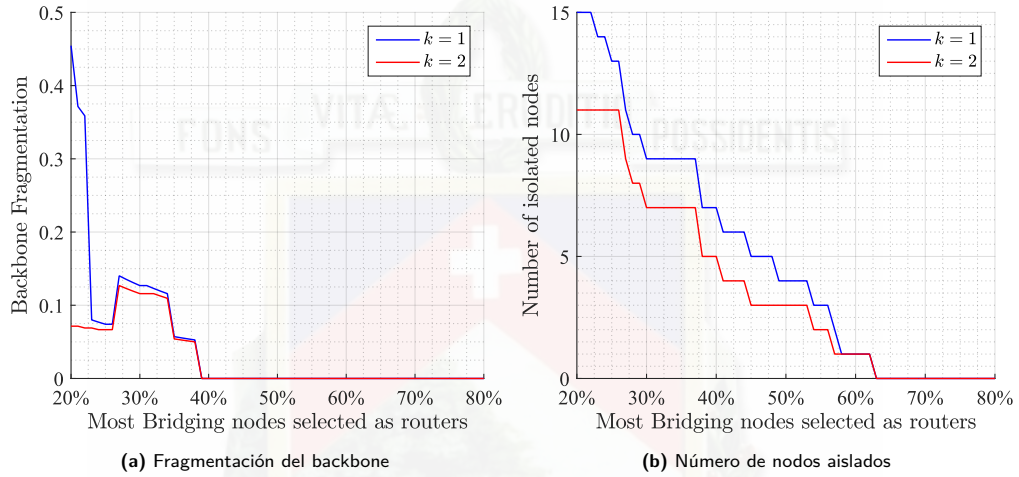


Figura 4.4: Método *k nodos más centrales en cada comunidad* para distintos valores de porcentaje de “nodos puente”, para el escenario SWIM en $t = 30$ seg

4.3.2. Vecino con mayor valor de Intra-Centralidad de la misma comunidad (C-A H_{IntraC_BN})

Evidentemente el número de nodos aislados es el principal inconveniente del método anterior, por lo que este y el siguiente método tratan de corregir este problema. Este método propuesto es denominado C-A H_{IntraC_BN} , correspondiente a las siglas en inglés de: *Community-Aware Highest Intra-Centrality Neighbor*. Consiste en que cada nodo de la red escoge a su vecino con mayor valor de $intraC_B$ como enrutador y adicionalmente se escoge un cierto porcentaje de nodos con mayor valor de centralidad de puentado C_{Brid} de toda la red al igual que el método anterior. Notar que según la Ec(2.8), $intraC_B$ se define como la centralidad betweenness de un nodo evaluado exclusivamente en su comunidad. Obviamente los nodos que no pertenezcan a la misma comunidad del nodo en cuestión, tienen $intraC_B = 0$, por lo que simplemente no se consideran. El hecho de que cada nodo escoja a un vecino inmediato como enrutador, garantiza que no queden nodos aislados en la red (a excepción de aquellos que de por sí, no tengan ningún enlace establecido). En la Fig. 4.5a se visualiza la fragmentación del backbone generado con este método con línea azul. Con este método, tan solo es necesario un 20 % de “nodos puente” para que la fragmentación sea igual a cero y que no existan nodos aislados (Fig. 4.5b).

4.3.3. Vecino con mayor valor de Centralidad Betweenness de la misma comunidad (C-A HC_BN)

En el presente método propuesto denominado $C-A HC_BN$ (siglas correspondientes de: *Community-Aware Highest Betweenness Centrality Neighbor*) a diferencia del anterior, cada nodo escoge a su vecino con mayor valor de C_B que pertenezca a la misma comunidad. Es decir, un nodo calcula su centralidad betweenness independientemente de la comunidad a la que pertenezca, o de las comunidades a las que pertenezcan el resto de los nodos. Pero solo son elegibles aquellos nodos vecinos que pertenezcan a la misma comunidad del nodo en cuestión. También se escoge adicionalmente un cierto porcentaje de nodos con mayor valor de centralidad de puentado C_{Brid} de toda la red como en los dos métodos anteriores. Este método trata de simplificar el hecho de que un nodo deba conocer su comunidad para poder calcular su valor de intra-centralidad. Se requiere conocer de la comunidad del nodo solamente para restringir a sus vecinos elegibles. En la Fig. 4.5a se visualiza la fragmentación del backbone generado con este método con línea roja. Tan solo es necesario un 20 % de “nodos puente” para que la fragmentación sea igual a cero y que no existan nodos aislados (Fig. 4.5b). Sin embargo, este método alcanza valores mucho menores de fragmentación de backbone que $C-A HIntraC_BN$ para un porcentaje menor de “nodos puente” seleccionados.

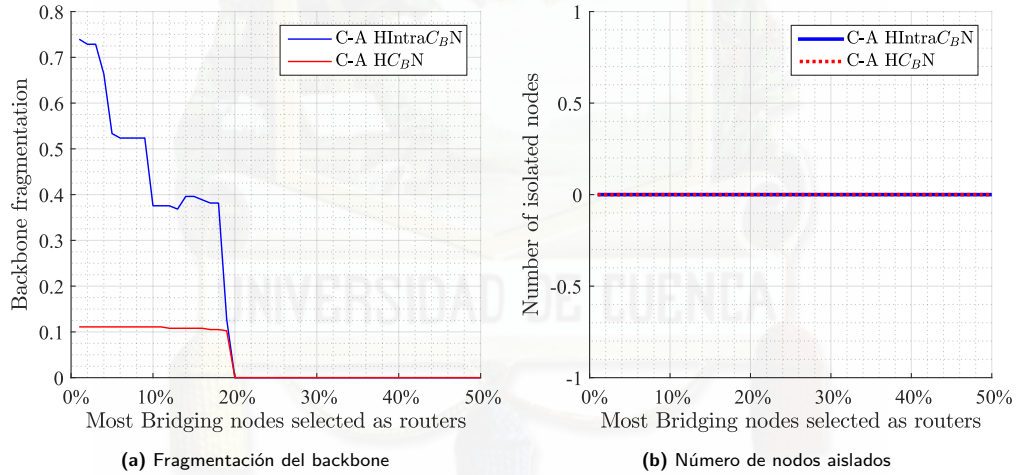


Figura 4.5: Métodos $C-A HIntraC_BN$ y $C-A HC_BN$ para distintos valores de porcentaje de “nodos puente”, para el escenario SWIM en $t = 30$ seg

En las Fig. 4.6a,4.6c,4.7a,4.8a se ilustran las topologías logradas con cada uno de los 3 métodos descritos. Los nodos de color azul representan aquellos escogidos como enrutadores MRs. Los nodos de color verde representan los nodos cliente MCs que solamente pueden comunicarse con sus vecinos enrutadores. Finalmente, los nodos de color rojo representan nodos aislados. En las Fig. 4.6b,4.6d,4.7b,4.8b en cambio, se muestran los backbones resultantes, es decir, las

topologías conformadas únicamente por MRs para cada caso correspondiente.

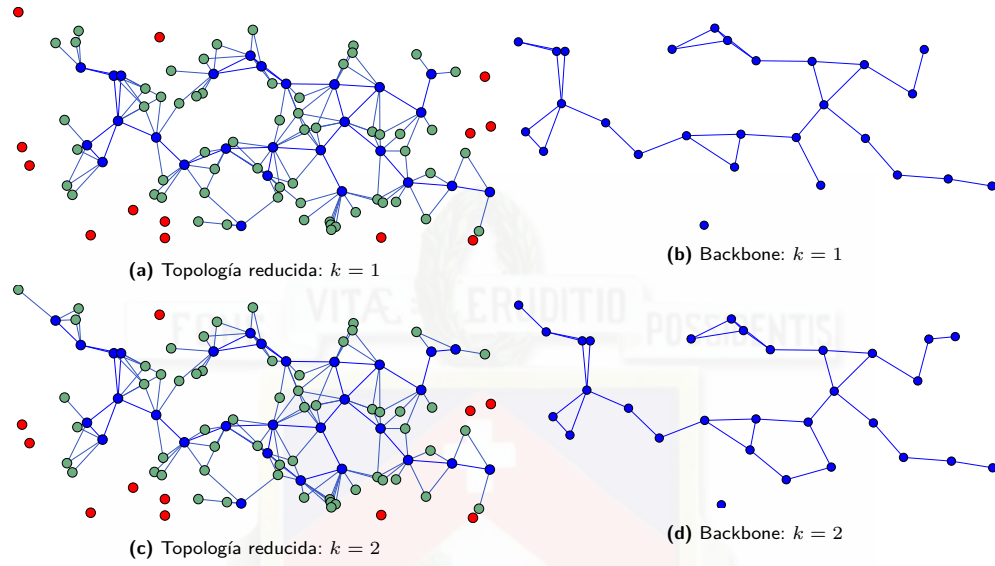


Figura 4.6: Topología reducida y backbone resultante con el método k nodos más centrales en cada comunidad con el 25 % de “nodos puente”, para el escenario con movilidad **SWIM** en $t = 30seg$

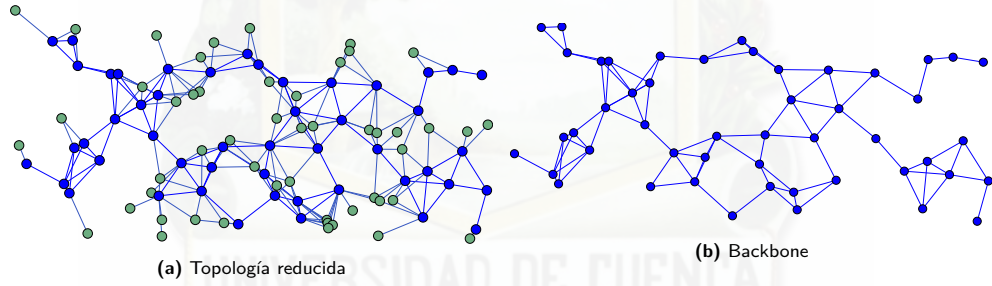


Figura 4.7: Topología reducida y backbone resultante con el método **C-A HIntraCBN** seleccionando el 25 % de “nodos puente” para el escenario con movilidad **SWIM** en $t = 30seg$

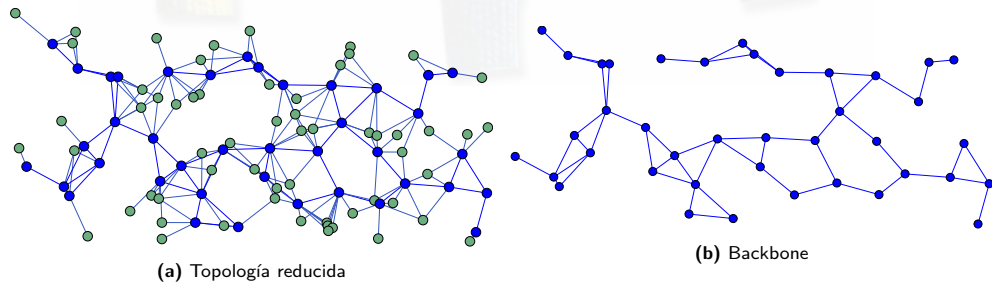


Figura 4.8: Topología reducida y backbone resultante con el método **C-A HCBN** seleccionando el 25 % de “nodos puente” para el escenario con movilidad **SWIM** en $t = 30seg$

En la Tabla 4.7 se resumen los porcentajes de “nodos puente” requeridos de tal modo que no exista fragmentación en el backbone y para que el número de nodos aislados sea cero para este caso analizado específico. En el caso de $C-A HC_B N$, tan solo se requieren de 39 de los 100 nodos que cumplan la función de enrutador. Nótese que 20 de éstos nodos (20 %) se han escogido debido a su valor de C_{Brid} mientras que los 19 restantes son escogidos exclusivamente por alguno/algunos de sus vecinos por tener el valor más alto de C_B dentro de los vecindarios de éstos.

Método	nodos puente	nodos por comunidad	Enrutadores seleccionados
k nodos más centrales en cada comunidad	63 %	$k = 1$	63
k nodos más centrales en cada comunidad	63 %	$k = 2$	63
C-A $HIntraC_B N$	20 %	-	46
C-A $HC_B N$	20 %	-	39

Tabla 4.7: Número de enrutadores seleccionados para valores mínimos de $F = 0$

Recordemos que estos resultados corresponden específicamente al escenario con modelo de movilidad $SWIM$ en la captura instantánea correspondiente al tiempo $t = 30seg$. Pero tras analizar varias capturas de todos los escenarios propuestos, los resultados tienen comportamientos similares. De manera general, concluimos que el método $C-A HC_B N$ logra los mejores resultados de conectividad de red. Sin embargo, es necesario definir un valor de porcentaje de “nodos puente”. En [70] definen como “nodos puente” aquellos que están dentro del 25 % con mayor valor de C_{Brid} en la red debido a estudios empíricos que condujeron a esta conclusión. Entonces para la presente propuesta de mecanismo de control de topología, se define el porcentaje de “nodos puente” en 25 %.

4.4. Análisis de topologías construidas

En este apartado se analizarán el número de enrutadores, los estados cambiantes en los nodos, y el número de aristas en la red para el mecanismo de control $C-A HC_B N$ que alcanzó mejores características de conectividad con un número reducido de MR seleccionados. Las simulaciones realizadas son de 500 segundos y se asume que el control de topología se actualiza cada 5 segundos, es decir, que cada 5 segundos se evalúa la topología y se realizan cambios en los estados de los nodos en caso de ser necesarios.

En la Tabla 4.8 se resume el número de nodos elegidos por el método $C-A HC_B N$ para que

cumplan la función de **MR** durante los 500 segundos de simulación para los seis modelos de movilidad estudiados. El valor promedio de enrutadores escogidos es de aproximadamente el 40 %. Eso significa que en todos los escenarios de movilidad es necesario menos de la mitad de nodos **MR** mientras que los demás únicamente son **MC**.

También es interesante notar que para el caso de movimiento aleatorio **RWM** de los nodos, se tiene la mayor cantidad de número de **MR** seleccionados. Esto confirma la hipótesis de que el método propuesto efectivamente funciona mejor para modelos de movilidad que presentan estructuras en comunidades tal como en escenarios presentes en la vida real.

Modelo	min	max	media	std
SWIM	34	48	40.28	2.9373
SLAW	31	44	36.34	2.3665
SMOOTH	34	49	41.43	2.8683
Disaster A.	31	44	36.33	2.7379
Geographic R	33	46	37.66	3.1502
RWM	38	49	42.19	2.5174

Tabla 4.8: Número de enrutadores seleccionados con el método **C-A HC_BN** para escenarios de 500 segundos

Por otra parte, en la Tabla 4.9 se resume el número de cambios de estado en los nodos durante 500 segundos de simulación de cada modelo de movilidad. Un nodo puede encontrarse en uno de 2 posibles estados: **MC** y **MR**. Un cambio de estado se define como la conmutación entre uno de éstos estados y otro. Y el número de cambios de estado de un nodo se define como el número de veces que un nodo tuvo que conmutar entre un estado u otro. Los histogramas completos de este parámetro para cada modelo de movilidad se ilustran en la Fig. 4.9.

Como era de esperarse, para el escenario de desastres y para el modelo **SWIM** es donde existen mayor cantidad de cambios de estado debido a la alta movilidad y dinamismo que existe ya que los nodos cambian de grupo o comunidad con mayor frecuencia. En dichos modelos se tiene una media de 23 cambios de estados en 500 segundos lo que equivale a una tasa de cambio de estado de 2.76 veces/minuto. El menor número de cambios es alcanzado por el modelo **SMOOTH** ya que los nodos tienden a moverse únicamente hacia vecindades familiares cercanas y como consecuencia los nodos tienen una tasa de cambio de estado media de 1.464 veces/minuto.

Los cambios de estado tienen un impacto directo en el consumo energético de los nodos. La asignación temporal de roles, permiten a los nodos apagar sus funcionalidades de enrutamiento de vez en cuando, evitando que se produzcan periodos muy largos de tiempo en el que puedan desgastar por completo sus baterías.

Modelo	min	max	media	std
SWIM	8	35	23.36	6.1406
SLAW	5	28	15.96	5.7943
SMOOTH	0	32	12.2	6.653
Disaster A.	6	40	22.77	7.5235
Geographic R	2	26	13.35	5.3435
RWM	0	32	15.78	7.6033

Tabla 4.9: Número de cambios de estado de los nodos con el método **C-A HC_BN** para escenarios de 500 segundos

Por último, en la Fig. 4.10, se grafican las curvas evolutivas en el tiempo del número de enlaces en la red. El número de enlaces activos disminuye con respecto a una topología en la que todos los nodos son enrutadores, al aplicar el mecanismo de control de topología, y por ende, se disminuyen también las colisiones y mensajes de enrutamiento en la red como veremos posteriormente en la evaluación del desempeño de la red.

La Tabla 4.10 resume el promedio de número de aristas obtenidos durante los 500 segundos de simulación para el caso de la red con y sin control de topología. Los porcentajes de la reducción de aristas logrados con el control de topología se resumen en la Tabla 4.11. La cantidad de aristas (enlaces) en la red reduce en un 30.52-43.97 % lo que ayuda a evitar interferencia entre distintos enlaces.

Modelo	All Routers	C-A HC _B N
SWIM	356.54	232.65
SLAW	541.14	320.31
SMOOTH	356.09	245.93
Disaster A.	486.65	284.31
Geographic R	480.03	267.76
RWM	336.03	224.11

Tabla 4.10: Número promedio de aristas en la topología durante 500 segundos de simulación.

Modelo	Porcentaje
SWIM	34.58 %
SLAW	40.77 %
SMOOTH	30.52 %
Disaster A.	41.47 %
Geographic R	43.97 %
RWM	33.26 %

Tabla 4.11: Porcentaje de disminución de aristas en la topología aplicando el método **C-A HC_BN** con respecto a topología completa durante 500 segundos de simulación.

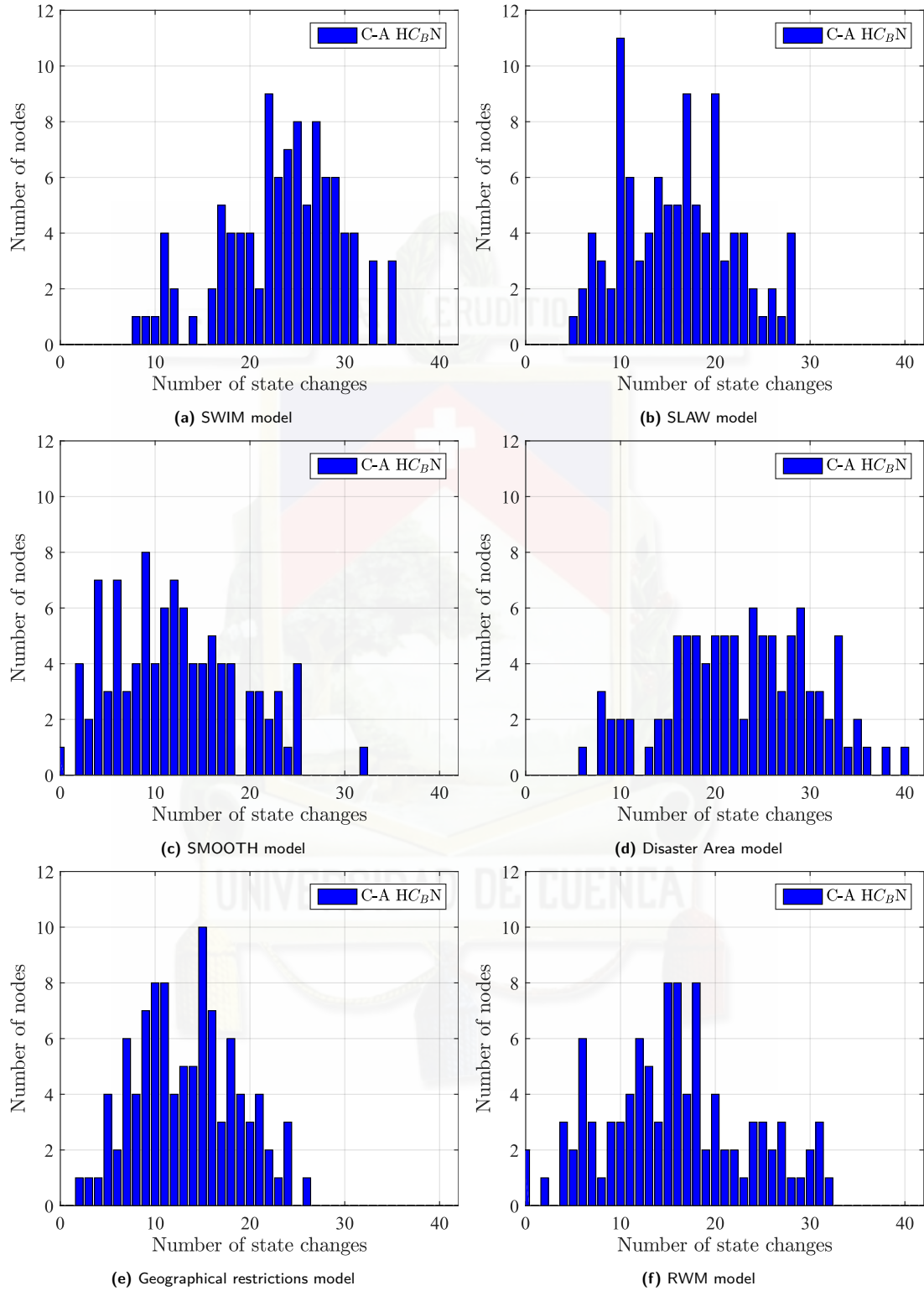


Figura 4.9: Histogramas de cambios de estado de los nodos

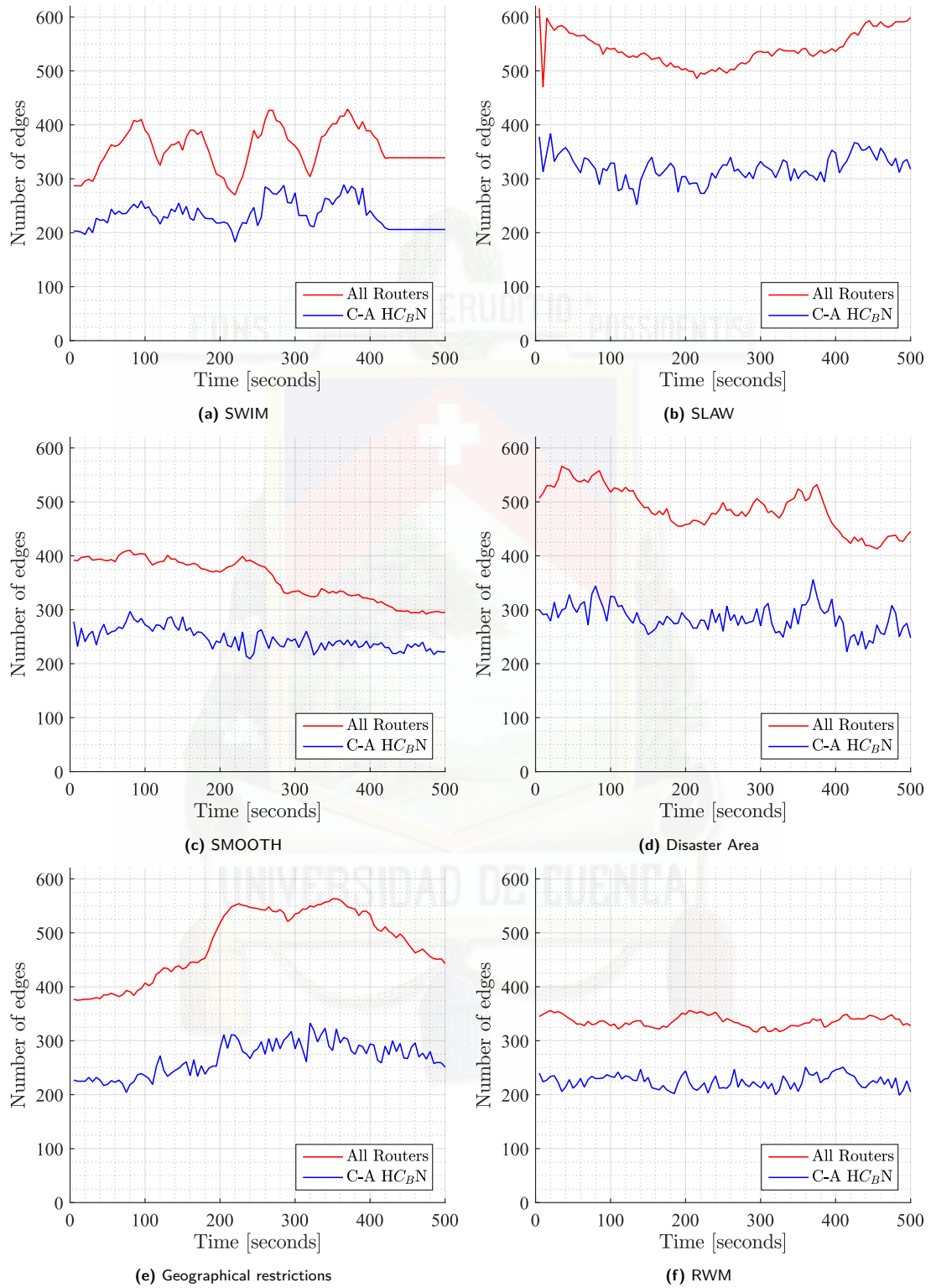


Figura 4.10: Evolución del número de aristas en la red a través de 500 segundos de simulación. Resultados para los seis distintos escenarios.

4.5. Conclusiones

En este capítulo se llevó a cabo el diseño de un mecanismo de control de topología para redes inalámbricas de tipo malla. En primer lugar se escogieron 6 distintos escenarios para la evaluación del mecanismo: [SWIM](#), [SLAW](#), SMOOTH, Área de desastres, Restricciones geográficas y [RWM](#).

La detección de comunidades se lleva a cabo con el algoritmo de *Louvain* de manera paralela mediante el software Grappolo [81], alcanzando modularidades mayores a 0.6 en promedio.

Se han probado distintos métodos de selección de enrutadores de los cuales el método [C-A HC_BN](#) consigue mejor conectividad de red con un número reducido de nodos [MR](#). Este método consiste en que cada nodo escoge como [MR](#) a su nodo vecino con mayor valor de centralidad *betweenness* C_B que pertenezca a su comunidad. Adicionalmente se escoge el 25 % de nodos en la red que tengan mayor valor de centralidad de puenteado C_{Brid} .

Con el método [C-A HC_BN](#) se logra obtener topologías reducidas con un porcentaje de nodos [MR](#) aproximado de 40 % para todos los escenarios. En los modelos de Área de desastre y [SWIM](#) los nodos tienen una tasa de cambio de estado promedio de 2.76 veces/minuto debido a la alta movilidad que se presentan en estos modelos. El caso de menor cambios de estado es alcanzado por SMOOTH en donde un nodo cambia de estado en promedio 1.464 veces/minuto. Estos cambios de estado tienen impacto positivo directo en el ahorro energético de los nodos.

Como consecuencia de la reducción de nodos [MR](#), el número de aristas en la red decrece entre un 30.52 % y 41.47 % en comparación a la topología completa en donde todos los nodos son enrutadores.



Capítulo 5

Metodología

En este capítulo presentamos la metodología que se llevó a cabo para el desarrollo de este trabajo.

Para explicación de la metodología del trabajo realizado dividimos el proceso completo en los siguientes pasos:

1. Generación de trazas de Movilidad: BonnMotion
2. Generación de trazas de movilidad: SUMO
3. Creación de la matriz de adyacencia
4. Detección de Comunidades
5. Selección de routers
6. Simulación de red con tráfico
7. Gráficos de topologías

5.1. Generación de trazas de movilidad: Bonnmotion

Bonnmotion es un simulador y generador de trazas de movilidad disponible en línea en [84]. Este simulador es comúnmente utilizado para el análisis de características de redes *ad hoc*. Las trazas de los modelos de movilidad: SWIM, SLAW, SMOOTH, Área de Desastre y RWM fueron obtenidas con este software. Las configuraciones para cada modelo se explicaron previamente en la Sección 4.1.

BonnMotion está basado en modelos de movilidad de licencia GNU. Estos modelos de movilidad son desarrollados por sus respectivos autores en distintos lenguajes de programación. Bonnmotion reúne y adapta todos estos modelos a lenguaje JAVA, con el propósito de simplificar y facilitar la configuración de los parámetros de cada modelo. Además, cuenta con una aplicación que ayuda a exportar los datos generados a formatos compatibles con simuladores de red como ns-2, ns-3, entre otros.

La información de instalación, configuración y uso del software se encuentran disponibles en [84]. Los modelos de movilidad compatibles y más información adicional se puede encontrar en el trabajo realizado por los desarrolladores del software en [85].

5.2. Generación de trazas de movilidad: SUMO

SUMO [26] (Simulation of Urban MObility), es un simulador de movilidad urbana basado en áreas realistas con el cual se puede generar trazas de movilidad basado en restricciones geográficas. Incluye varias aplicaciones que permiten preparar y realizar simulaciones de tráfico (pedestre y vehicular) al incluir mapas OSM con lo cual puede crear regiones donde los nodos puedan movilizarse. Este simulador es de distribución GNU y desarrollado en lenguaje C++.



SUMO es especializado en la generación de rutas para vehículos. Sin embargo, para este trabajo utilizamos las aplicaciones que permiten generar movimientos pedestres. Requiere un mapa **OSM** obtenido de [86].

Al mapa resultante con formato compatible se le importan polígonos a los puntos de interés para tener una mejor apreciación del mapa en el entorno gráfico de **SUMO**. Esto se realiza mediante el comando **polyconvert** que importa las formas geométricas de acuerdo a los contornos y colores de: calles, veredas, construcciones, parques, etc. Esta aplicación requiere diferentes fuentes como: **-osm-files** que ingresa el mapa **OSM** para tener las posiciones, **-net-file** que ingresa la red anteriormente generada para tener referencias de desplazamientos y proyecciones, **-type-file** ingresa un archivo que contiene la configuración con el tipo de polígonos, identificador de regiones, colores, capas, entre otros (configuración disponible por los desarrolladores en [86]). La aplicación genera como salida un único archivo XML que se obtiene mediante el comando **-o**, seguido del nombre del archivo.

Con el entorno configurado, se procede con la generación de los viajes aleatorios mediante la aplicación **randomTrips.py**. La aplicación tiene como uno de los parámetros de entrada el archivo de la red generado por **netconvert**. Otro parámetro de entrada es la hora de inicio y fin (en segundos) del tráfico, con la opción **-b** y **-e** respectivamente. Para el enfoque de este trabajo se realiza una configuración particular mediante la opción **-pedestrian** para movimientos pedestres. Por último es necesario especificar la máxima distancia que recorren los nodos mediante **-max-dist**.

Esta aplicación arroja como resultado un archivo XML para **DUAROUTER** utilizando la opción **-r**, el cual sirve para establecer y calcular la ruta más corta de vehículos y personas a partir de diferentes demandas. En este trabajo, la aplicación se configura inicialmente indicando que se trata de movimientos de tipo pedestre con la opción **-c** seguido de **pedestrian**, luego con la opción **-t** para leer las rutas mediante un identificador pedestre: **"type=\"ped\""**.

Para finalizar, es necesario exportar los viajes de los nodos a una traza compatible con ns-2. **SUMO** mediante la aplicación **traceExporter.py** permite la exportación de trazas en diferentes formatos. Para este trabajo las opciones de salida requeridas son: **-ns2mobility-output** y **-ns2config-output**, para obtener la traza y la configuración (tiempo de simulación, nodos involucrados y área de simulación) respectivamente.

5.3. Creación de la matriz de adyacencia

Una vez creado el archivo de traza de movimientos en formato ns2, nos ayudamos del *Ns2MobilityHelper* de ns-3 para importarlos e instalarlos en los nodos mediante las líneas:



```
Ns2MobilityHelper ns2 = Ns2MobilityHelper (traceFile);
nodes.Create (nodeName);
ns2.Install ();
```

donde `traceFile` es la variable con el nombre del archivo de trazas y `nodeName` es el número de nodos.

Entonces, procedemos a evaluar los estados de enlace entre los nodos tomando en cuenta que sus posiciones cambian dinámicamente siguiendo un modelo de movilidad. Para ello generamos dos tipos de reportes cada 5 segundos. `Report()` recopila datos de interfaz, [HWMP](#), y Peer Management Protocol descritos en la en la Sección 2.1.1. Este último nos proporciona información acerca de los vecinos de un nodo en específico, con lo cual podemos conocer las conexiones establecidas en la red. `ReportPos()` por su parte genera reportes con las posiciones exactas de los nodos en intervalos de 5 segundos para posteriormente poder generar un gráfico de la red.

```
void
MeshTest::Report ()
{
    unsigned n (0);
    for (NetDeviceContainer::Iterator i = meshDevices.Begin (); i != ...
        meshDevices.End (); ++i, ++n)
    {
        std::ostringstream os;
        os << outFolder << "/MeshReports/Sec" << Simulator::Now().GetSeconds() ...
        << "/mp100AllR-report-" << n << ".xml";
        std::ofstream of;
        of.open (os.str ().c_str ());
        if (!of.is_open ())
        {
            std::cerr << "Error: Can't open file " << os.str () << "\n";
            return;
        }
        mesh.Report (*i, of);
        of.close ();
    }
    std::cerr << "Printing mesh point devices at time #" << ...
    Simulator::Now().GetSeconds() << "\n";
}
```

```
void
MeshTest::ReportPos ()
{
    std::ostringstream mystring;
    mystring << outFolder << "/MeshPositions/Sec" << Simulator::Now().GetSeconds() ...
    << ".txt";
}
```



```
std::ofstream myfile;
myfile.open (mystring.str ().c_str ());
myfile <<"X_Pos      Y_pos " << nodeNum << std::endl;
for (int i = 0; i < nodeNum; i++)
{
    Ptr<MobilityModel> mob = nodes.Get(i)->GetObject<MobilityModel>();
    Vector pos = mob->GetPosition ();
    myfile << pos.x <<" " <<pos.y<< std::endl;
}
myfile.close();
}
```

A continuación se presenta una sección de un reporte mesh generado con `Report()`. Podemos ver que se trata del nodo número 4 (address="00:00:00:00:00:04") en algún instante de tiempo específico. Según las estadísticas de *Peer Management Protocol*, este nodo tiene 2 vecinos activos (linksTotal="2"). Los vecinos son los nodos 72 y 31 (48 y 1F en hexadecimal) con métricas 150 (metric="150") en ambos casos. Recordar que la métrica está definida en la Ec(2.1).

```
<PeerManagementProtocol>
<Statistics linksTotal="2" linksOpened="2" linksClosed="0"/>
<PeerManagementProtocolMac address="00:00:00:00:00:04">
<Statistics txOpen="2"
txConfirm="2"
txClose="0"
rxOpen="2"
rxConfirm="2"
rxClose="0"
dropped="0"
brokenMgt="0"
txMgt="4"
txMgtBytes="132"
rxMgt="4"
rxMgtBytes="124"
beaconShift="1"/>
</PeerManagementProtocolMac>
<PeerLink
localAddress="00:00:00:00:00:04"
peerInterfaceAddress="00:00:00:00:00:48"
peerMeshPointAddress="00:00:00:00:00:48"
metric="150"
lastBeacon="3.1927"
localLinkId="1"
peerLinkId="1"
assocId="0"
/>
<PeerLink
localAddress="00:00:00:00:00:04"
```



```
peerInterfaceAddress="00:00:00:00:00:1f"  
peerMeshPointAddress="00:00:00:00:00:1f"  
metric="150"  
lastBeacon="0.346207"  
localLinkId="2"  
peerLinkId="3"  
assocId="1"  
/>  
</PeerManagementProtocol>
```

Una vez obtenidos los reportes, creamos un script en Octave [28] para extraer los vecinos de todos los nodos con sus métricas. Con ello creamos las matrices de conectividad y de métricas de la red entera en cada instante de tiempo (5 segundos), así como una matriz con las coordenadas de los nodos. Debido a que las simulaciones constan de 100 nodos cada una, las matrices de adyacencia y métricas son de 100×100 en donde las filas corresponden al nodo origen y las columnas corresponden al destino.

Estas matrices definen completamente a la topología de la red por lo que ahora ya podemos realizar procesamiento y control de topología con estas matrices cargadas en Octave. El primer paso es pasar la matriz de métricas a un formato admisible para la detección de comunidades que explicaremos en la siguiente sección.

5.4. Detección de Comunidades

Una vez que tenemos descrita nuestro grafo completo mediante la matriz de métricas, podemos realizar la detección de comunidades. Para ello hacemos uso del software gratuito Grappolo [81]. Grappolo es un programa basado en C++ que implementa el algoritmo de Louvain [80] de manera paralela para la detección de comunidades como se explica en la Sección 4.2.

Grappolo admite 6 tipos de formato de archivo como entrada: Matrix-Market, DIMACS#9, Pajek, Metis, Simple edge list y Binary format. Sin embargo el mejor soporte es para archivos de formato Metis. Incluso el ejemplo proporcionado en el programan, tiene este formato. A continuación explicamos el formato de un archivo Metis con un ejemplo [87]:

```
7 11 11
4 5 1 3 2 2 1
2 1 1 3 2 4 1
5 5 3 4 2 2 2 1 2
3 2 1 3 2 6 2 7 5
1 1 1 3 3 6 2
6 5 2 4 2 7 6
2 6 6 4 5
```

La primera línea corresponde al encabezado del archivo y especifica: el número de nodos (7), el número de aristas (11), y el valor de FMT (11). FMT tiene los siguientes significados:

- 0 el grafo no es ponderado (se puede omitir el FMT);
- 1 el grafo tiene pesos en las aristas;
- 10 el grafo tiene pesos en los vértices;
- 11 el grafo tiene pesos en las aristas y en los vértices.

Las siguiente línea se lee como sigue: el nodo 1 tiene un peso 4 y se conecta al nodo 5 con un peso de enlace 1, al nodo 3 con un peso de enlace de 2 y al nodo 2 con un peso de enlace de 1.

Una vez que tenemos los archivos de entrada, nos ubicamos en el directorio de Grappolo y procedemos a ejecutar la utilidad de detección de comunidades con el siguiente comando:

```
\$ driverForGraphClustering Graph_Sec_5.graph -f 5 -o
```

Donde **Graph_Sec_5.graph** es el nombre del archivo en formato Metis que contiene nuestra topología completa, **-f 5** especifica que el formato del archivo de entrada es Metis, y **-o** habilita la creación del archivo de salida. La salida de la línea de comandos es la siguiente:

```
***** Compact Summary *****
*****
Number of threads           : 4
Total number of phases     : 4
Total number of iterations  : 18
Final number of clusters   : 10
Final modularity            : 0.771501
Total time for clustering   : 0.000942
Total time for building phases : 0.000527
*****
TOTAL TIME                  : 0.001469
*****
```

Esta salida indica que se han detectado 10 comunidades y la modularidad alcanzada es de 0.771501. También se genera el archivo de salida `Graph_Sec_5.graph_clustInfo` el cual contiene un vector con el número de comunidad correspondiente a cada uno de los nodos de la red. Este vector es finalmente importado a Octave para seguir con el el respectivo control de topología.

5.5. Selección de Enrutadores

El control de topología se basa en la selección de nodos que cumplan el rol de enrutadores. Hasta este punto tenemos la matriz de adyacencia/conectividad, la matriz de métricas, las coordenadas de los nodos, y las comunidades a las que pertenecen los nodos. Procedemos entonces a crear un script en Octave [28] que procese las matrices para el control de topología. El script principal se aprecia en el Apéndice B.1. Las subfunciones que fueron creadas son las siguientes:

- `Adjacent_Matrix`: Genera los archivos (.txt) con la matrices de adyacencia a partir de los reportes generados por ns3. También genera archivos con las matrices de métricas.
- `write_matrix_to_metis`: Crea los archivos de las topologías en un formato Metis (.graph). Estos archivos se obtienen a partir de las matrices de métricas.
- `DetectCom`: Llama a la función `system()` que ejecuta la detección de comunidades desde la línea de comandos de Linux.
- `betweenness_bin`: función para calcular los valores de centralidad de los nodos [88].
- `MostCentralCom`: Devuelve un vector con los dos nodos más centrales elegidos como routers.
- `ReducedTopologyBridging`: Devuelve el vector R con los nodos seleccionados para ser routers mediante el método *C-A HC_BN* propuesto en la Sección 4.3.3
- `ReducedTopologyBackbone`: Devuelve el vector R con los nodos seleccionados para ser routers mediante el método de árbol de expansión explicado en la Sección 7

5.6. Simulación de red con tráfico

Finalmente, debemos evaluar nuestro método de control de topología enviando tráfico en redes. Para ello, hacemos nuevamente uso del simulador de redes ns-3. **Network Simulator 3** es un simulador de redes de eventos discretos basado en C++, es decir que simula eventos con *timestamp* de tal manera que el reloj interno del simulador avanza de manera discreta. En cuanto a redes mesh concierne, ns-3 es la opción preferida para investigaciones académicas, según [89]. Es un programa gratuito bajo la licencia GNU GPLv2 [90].

En [32] se presenta el modelo de redes en malla inalámbricas implementado en ns-3. El modelo está localizado en `src/devices/mesh` dentro del directorio de ns-3.

En el Anexo B.2 se puede apreciar el código principal de ns-3 para llevar a cabo las simulaciones de red con tráfico. La variable `Routers` es un vector de ceros y unos de tamaño 100. Cada elemento en `Routers` determina si el nodo correspondiente a esa posición es un enrutador o no. En el caso de ejemplo ilustrado, todos los nodos son enrutadores. Para apagar las funcionalidades de enrutamiento en los nodos, modificamos el protocolo HWMP en ns-3. En el archivo `HwmpProtocol` agregamos el parámetro `IsRouter` el cual determinará si un nodo enviará o no mensajes de petición de ruta `PREQ`. `traceFile` es la variable en donde se almacena la traza de movilidad en formato ns-2 obtenida de BonnMotion o SUMO. En el archivo `TrafficDirectory.cc` se encuentra la programación del tráfico UDP a enviar. A continuación vemos un ejemplo de código para enviar tráfico desde el nodo 7 (cliente) hasta el nodo 8 (servidor):

```
//Server Side
UdpServerHelper server_7_8 (4015);
ApplicationContainer Sapps_7_8 = server_7_8.Install (nodes.Get(8));
Sapps_7_8.Start (Seconds (40.0));
Sapps_7_8.Stop (Seconds (m_totalTime-5));

//Client Side
UdpClientHelper udpClient_7_8 (interfaces.GetAddress (8), 4015);
udpClient_7_8.SetAttribute ("MaxPackets", UintegerValue (m_maxPacketCount));
udpClient_7_8.SetAttribute ("Interval", TimeValue (Seconds (m_packetInterval)));
udpClient_7_8.SetAttribute ("PacketSize", UintegerValue (m_packetSize));
ApplicationContainer Capps_7_8 = udpClient_7_8.Install (nodes.Get(7)) ;
Capps_7_8.Start (Seconds (50.0));
Capps_7_8.Stop (Seconds (m_totalTime-50));
```

En el lado del servidor, se establece el puerto, nodo, tiempo de inicio, y tiempo de fin de la aplicación del servidor. Mientras tanto, en el lado cliente se establece puerto, nodo, máximo de paquetes a transmitir, intervalo de envío de paquetes, tamaño de paquete, tiempo de inicio y tiempo de fin de aplicación del cliente.

Los archivos generados son, el monitor de flujo y los reportes *mesh reports*. Al final obtendremos los datos de estos archivos de salida para proyectar los resultados de desempeño de la red.

5.6.1. Gestión de las simulaciones

`ns-3` se divide en un simulador de *core* y una parte de modelos. La parte del simulador es únicamente un simulador de eventos discretos, al cual el resto del código le brinda eventos para que los procese en un tiempo dado [91]. Ahora bien, las simulaciones realizadas, tardan un tiempo considerado en ser completadas y `ns-3` utiliza un solo núcleo del computador por simulación, dejando gran porcentaje de la capacidad del procesador desocupado. Para optimizar el tiempo de simulaciones, se ha decidido llevar a cabo distintas simulaciones en distintos núcleos del computador de manera paralela. Esto se puede lograr gracias al siguiente comando (Linux):

```
taskset -c <core> <task>
```

donde `<core>` es el número de núcleo al que queremos asignar nuestra tarea y `<task>` en nuestro caso, es la simulación respectiva que deseamos ejecutar. En la Fig. 5.1 podemos ver el resultado de ejecutar el comando `gnome-system-monitor` en el cual podemos observar el comportamiento de cada núcleo. En este ejemplo contamos con 4 núcleos. En la Fig. 5.1c vemos que se llevan a cabo 4 simulaciones, una en cada núcleo, lo que nos permite sacar provecho de toda la capacidad del computador y lograr agendar las simulaciones de mejor manera.

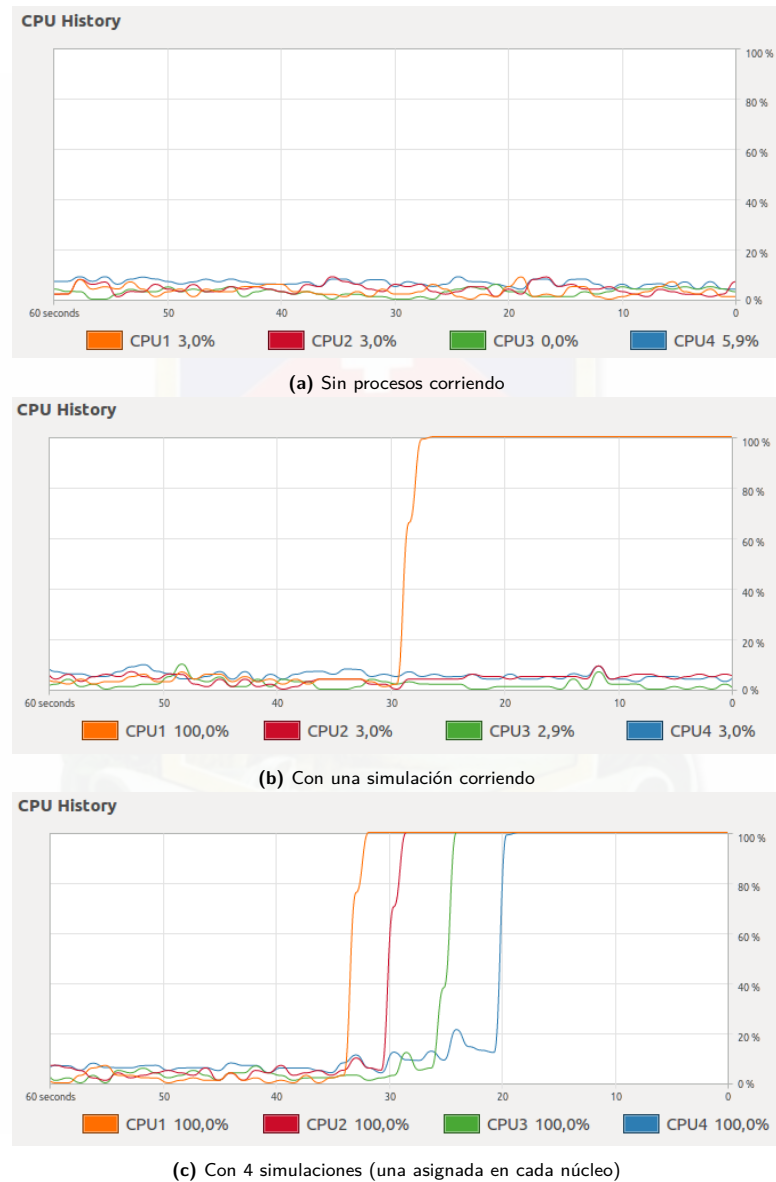


Figura 5.1: Comportamiento de los 4 núcleos del computador al correr simulaciones

5.7. Gráficos de topologías

Para los gráficos de las topologías, hemos utilizado el programa Gephi [27]. Gephi es un programa gratuito de licencia libre que puede correr en Windows, Mac OS, y Linux. Los formatos de archivos que soporta y sus principales características se presentan en la Tabla 5.1.

	Edge List/Matrix Structure	XML Structure	Edge Weight	Attributes	Visualization Attributes	Attribute Default Value	Hierarchical Graphs	Dynamics
CSV	X		X					
DL Ucinet	X		X	X				
DOT Graphviz			X		X			
GDF			X	X	X	X		
GEXF		X	X	X	X	X	X	X
GML			X	X	X			
GraphML		X	X	X	X	X	X	
NET Pajek	X		X		X			
TPL Tulip								
VNA Netdraw			X	X				
Spreadsheet			X	X				X

Tabla 5.1: Comparativa de formatos soportados por Gephi

El formato de archivos que elegimos es Pajek (extensión .net) debido a su simplicidad y a que cumple con los requerimientos para poder plasmar gráficamente de manera clara nuestros grafos. La sintaxis para el formato Pajek que hemos utilizado es el siguiente:

Pajek Syntaxis

```

*Vertices N
1 "vertice_label" coordX coordY coordZ value ic color
2 "vertice_label" coordX coordY coordZ value ic color
3 "vertice_label" coordX coordY coordZ value ic color
...
N "lab" coordX coordY value ic color
*Edges
init_vertex term_vertex value

```

Ejemplo de un grafo no ponderado (Fig. 5.2):

```
----- Pajek file example -----
*Vertices 5
1 "Node_1" 582.645000 328.499000 0.000000 ic yellow
2 "Node_2" 890.158000 459.549000 0.000000 ic green
3 "Node_3" 389.936000 437.936000 0.000000 ic red
4 "Node_4" 122.513000 337.825000 0.000000 ic orange
5 "Node_5" 256.273000 327.631000 0.000000 ic maroon
*Edges
1 3
1 4
2 5
3 4
4 5
5 2
```

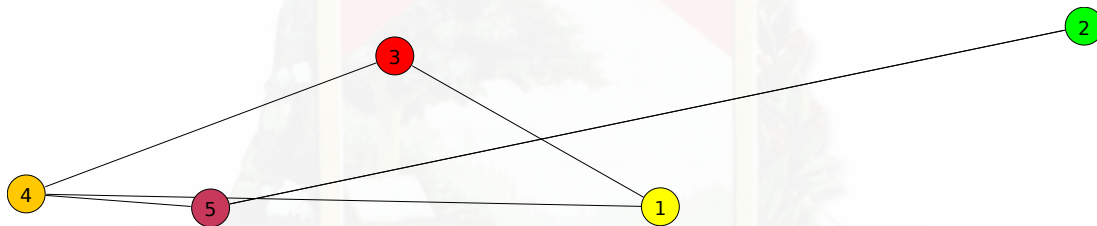


Figura 5.2: Pajek file example output





Capítulo 6

Resultados

En este capítulo se evalúa el desempeño de la red con y sin control de topología. En todos los casos se considera tráfico en tiempo real [UDP](#). Se comparan los resultados de la propuesta con aquellos obtenidos sin control de topología, y con redes con otras propuestas de control de topología que no toman en cuenta la estructura de comunidades.

En un trabajo previo [77] se propone un método de control de topología para WMNs. Dicho mecanismo se basa en métricas de centralidad y no toma en cuenta los pesos en las aristas/enlaces entre los nodos. El método consiste en que cada nodo escoge como MR a sus dos vecinos con mayor valor de centralidad betweenness. Se ha nombrado a este método como $2HC_BN$, siglas correspondientes de: *2 Highest Betweenness Centrality Neighbors*. Es un método descentralizado que muestra una mejora en el desempeño de la red y logra una mayor eficiencia energética en comparación a WMNs sin control de topología.

Al igual que C-A HC_BN , $2HC_BN$ también se basa en la selección de un CDS cuyos elementos conforman el backbone de la red. Sin embargo, este método anterior, no considera la estructura en comunidades de la red. Los autores evalúan $2HC_BN$ sobre una red cuyos nodos tienen un movimiento aleatorio. Pero, en la vida real, los dispositivos (transportados por personas) se movilizan de tal manera que socializan con otros dispositivos formando comunidades con sus vecinos. El objetivo principal de este apartado es el de medir el impacto en el desempeño de la red, al considerar la estructura comunitaria en el mecanismo de control de topología.

En las evaluaciones se considera principalmente el modo reactivo del protocolo de enrutamiento HWMP. Recordar que, el modo reactivo es comúnmente preferido en redes donde existe una alta movilidad de nodos y comunicaciones P2P (Sección 2.1.1.3). En este modo, la ruta se descubre únicamente cuando un paquete necesita ser transmitido desde un nodo fuente hacia un nodo destino, es decir, operación bajo demanda.

Para la evaluación un extenso conjunto de simulaciones fueron llevadas a cabo mediante el simulador de redes ns-3. Los 6 escenarios en los que se evalúa la red constan de 100 nodos que se mueven acorde a un modelo de movilidad distinto dentro de un área de 1024×512 m. Estos escenarios están descritos a detalle en la Sección 4.1. Para cada escenario se consideran 4 capturas instantáneas en las que se simulan de manera independiente, distintos números de flujos UDP (de 10 a 40 en intervalos de 5). Con el fin de obtener resultados confiables, para cada caso de número de flujos, se llevan a cabo 50 ejecuciones estadísticamente independientes. Al final se promedian los 200 resultados de cada caso. Los promedios se muestran con un intervalo de confianza del 95 % para todos los resultados.

Todos los flujos UDP utilizan una longitud de paquete de 512 bytes. El intervalo de envío de paquetes es de 20 ms. El tiempo total de transmisión de paquetes es de 200 segundos. Como resultado, 10000 paquetes son enviados en cada flujo P2P. Las características de radio frecuencia de los nodos están detalladas en la Tabla 6.1.

Los siguientes parámetros son evaluados: tasa de mensajes para gestión de enrutamiento, total de datos reenviados, número total de paquetes recibidos con éxito, número total datos reenviados por paquetes recibidos con éxito, eficiencia de la red en términos de proporción de entrega de paquetes (*packet delivery ratio* PDR), energía consumida por nodos, y el retardo de transmisión



Parámetro	Valor
Sensibilidad	-96 dbm
Ganancia Tx	1 dB
Ganancia Rx	1 dB
Potencia Tx	16.0206 dbm
Figura de Ruido Rx	7 dB
Frecuencia de Operación	2417 MHz
Ancho de canal	20 MHz

Tabla 6.1: Parámetros RF de los nodos

de paquetes extremo a extremo.

6.1. Impacto de la carga de tráfico, en redes con control de topología

Antes de comparar los métodos $C-A HC_{BN}$ y $2HC_{BN}$, se evalúa el beneficio de aplicar un mecanismo de control de topología en una WMN. Aunque el análisis se ha llevado a cabo con los seis modelos de movilidad evaluados, por motivos de ilustración, se presentarán solamente los resultados correspondientes al escenario con modelo de movilidad SWIM. En este escenario se evalúan los dos mecanismos de control de topología mencionados anteriormente, así como la red sin control de topología, donde todos los nodos cumplen la función de MR (*All Routers*).

Los resultados obtenidos en general se muestran en la Fig. 6.1. En ésta se observa claramente que la implementanci3n de un mecanismo de control de topología mejora el desempeño de la red en todos los aspectos: reducci3n de paquetes de enrutamiento (Fig. 6.1a), menor cantidad de reenvío de paquetes (Fig. 6.1b) y mayor número de paquetes recibidos (Fig. 6.1c). Consecuentemente la eficiencia de la red (Fig. 6.1e) aumenta en términos de proporci3n de entrega de paquetes. Por otro lado, al existir menos mensajes de enrutamiento y menor reenvío de paquetes por paquete recibido con éxito, se produce un menor consumo energético (Fig. 6.1f).

En la Tabla 6.2 se resumen los porcentajes de mejora de la red con los controles de topología $2HC_{BN}$ y $C-A HC_{BN}$, con respecto al escenario con todos los nodos actuando como enrutadores (*All Routers*). Se considera como mejora en el desempeño de la red, el porcentaje de:

- Disminuci3n de mensajes de enrutamiento
- Disminuci3n de reenvío de datos
- Aumento de paquetes recibidos
- Disminuci3n de paquetes reenviados
- Aumento en la eficiencia de la red
- Disminuci3n de consumo energético

La Tabla 6.2 permite concluir que son evidentes los beneficios de implementar un mecanismo de control de topología en una red. Se producen mejoras en todos los aspectos, reduciendo así problemas comunes como la colisi3n de paquetes y las interferencias generadas por la alta densidad de nodos.

A partir de este punto, este trabajo se centra en la comparaci3n individual de las métricas, entre los métodos $2HC_{BN}$ y $C-A HC_{BN}$ con el fin de determinar el impacto que produce el considerar la estructura en comunidades en el mecanismo de control de topología.

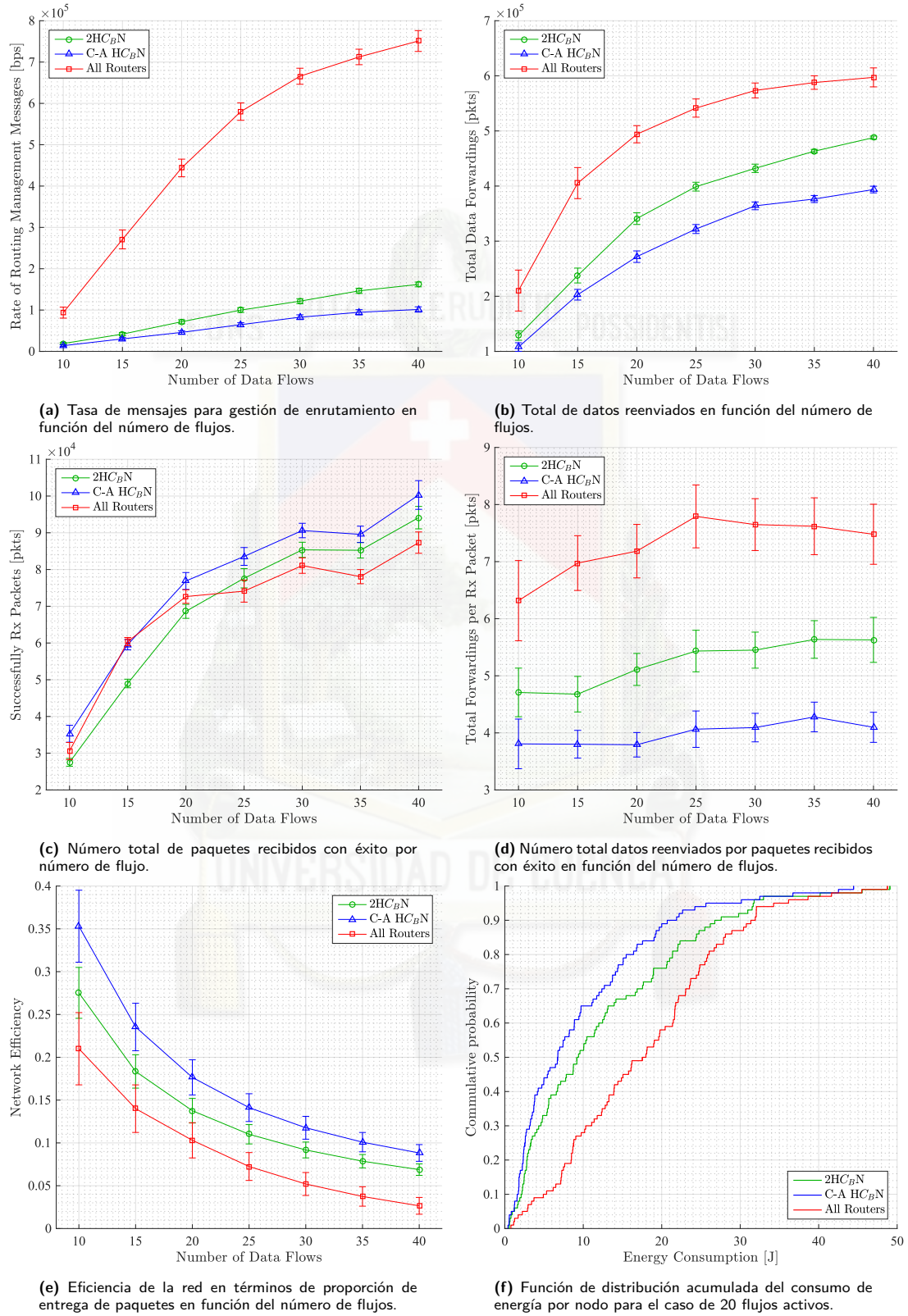


Figura 6.1: Resultados obtenidos de *All Routers* en comparación con los métodos de control de topología *2HC_BN* y *C-A HC_BN*, para el escenario con modelo de movilidad *SWIM*.

Parameter	Model	$2HC_BN$	C-A HC_BN
Rate of routing managemnet messages reduction	SWIM	81.50 %	87.63 %
	SLAW	83.14 %	86.55 %
	SMOOTH	75.88 %	82.69 %
	Disaster area	83.06 %	86.43 %
	Geographic Rest.	85.72 %	88.10 %
	RWM	73.44 %	80.64 %
Total data forwardings reduction	SWIM	28.81 %	41.53 %
	SLAW	15.42 %	27.11 %
	SMOOTH	13.48 %	24.22 %
	Disaster area	14.28 %	23.39 %
	Geographic Rest.	16.60 %	29.59 %
	RWM	13.82 %	28.97 %
Successfully Rx packets increase	SWIM	2.17 %	9.29 %
	SLAW	5.81 %	13.09 %
	SMOOTH	3.70 %	4.44 %
	Disaster area	8.40 %	7.11 %
	Geographic Rest.	8.74 %	10.36 %
	RWM	1.48 %	12.03 %
Total forwardings per Rx packet reduction	SWIM	28.15 %	45.12 %
	SLAW	23.70 %	31.20 %
	SMOOTH	17.55 %	29.55 %
	Disaster area	22.83 %	29.30 %
	Geographic Rest.	22.19 %	33.00 %
	RWM	13.31 %	27.91 %
Network Efficiency increase	SWIM	5.21 %	13.30 %
	SLAW	10.02 %	21.93 %
	SMOOTH	5.65 %	11.75 %
	Disaster area	3.87 %	10.95 %
	Geographic Rest.	4.87 %	13.29 %
	RWM	3.37 %	12.80 %
Energy consumption reduction	SWIM	30.52 %	46.18 %
	SLAW	12.53 %	24.72 %
	SMOOTH	11.17 %	26.44 %
	Disaster area	18.78 %	30.71 %
	Geographic Rest.	26.48 %	52.50 %
	RWM	8.49 %	31.36 %

Tabla 6.2: Resumen de porcentajes promedio de mejora de los mecanismos de control $2HC_BN$ y C-A HC_BN con respecto a la red operando con todos los nodos como enrutadores, para cada parámetros y modelo de movilidad.

6.1.1. Tasa media de mensajes para la administración del enrutamiento

La tasa media de mensajes para la administración de enrutamiento por flujo activo, para todos los modelos de movilidad considerados, se muestra en la Fig. 6.2. De dicha figura, es evidente que al tener una menor cantidad de estaciones seleccionadas como MR en el método C-A HC_BN, se ocasiona una menor tasa de mensajes de enrutamiento en la resolución de rutas. Como era de esperarse, al aumentar la carga de tráfico (la cantidad de flujos de datos), aumenta la tasa de mensajes de enrutamiento debido a que se está evaluando el protocolo de enrutamiento en modo reactivo (bajo demanda). Cabe destacar que esta reducción también impacta directamente en la energía disipada por los nodos. Esto debido a que aquellos nodos que no han sido seleccionados como enrutadores, no transmiten mensajes de administración de enrutamiento.

Para el modelo de movilidad con restricciones geográficas (Fig. 6.2e), se obtiene una tasa de mensajes de enrutamiento relativamente baja con ambos mecanismos de control de topología (menor a 60000 paquetes) en comparación a los otros modelos. Esto se debe principalmente a la disposición geográfica de los nodos. Esto es explicado a continuación mediante un ejemplo. En la Fig. 6.3a se puede apreciar una topología generada por 14 nodos que se ubican de acuerdo a una restricción geográfica. Debido a que los nodos se mueven únicamente por caminos (calles y veredas) rectos, no se producen conglomeraciones de nodos en ciertos sectores como llegaría a suceder en otros lugares como parques, ferias, centros comerciales, etc. Para este ejemplo, la topología creada tiene 14 aristas. Por otro lado, en la Fig. 6.3b se ilustra una topología creada por la misma cantidad de nodos pero sin restricciones geográficas. Los nodos son libres de movilizarse por toda el área, y debido al comportamiento social que influye en sus movimientos, éstos forman agrupaciones o comunidades en donde existe una alta densidad de enlaces. Para este caso específico, el número de aristas en la topología aumenta a 21. Esta diferencia en el número de enlaces, significa que existe menor cantidad de rutas posibles en escenarios con restricciones geográficas, lo que finalmente implica una menor cantidad de mensajes de enrutamiento.

En la Tabla 6.3 se resumen los porcentajes promedios de reducción de mensajes de enrutamiento del método propuesto C-A HC_BN con respecto a la propuesta anterior 2HC_BN. Se puede observar una mejora que va desde el 17 % para el modelo de movilidad con restricciones geográficas, hasta el 38.29 % para el modelo de movilidad SWIM.

SWIM	SLAW	SMOOTH	Disaster Area	Geographic Restrictions	RWM
38.29 %	20.55 %	28.35 %	19.81 %	17.00 %	27.19 %

Tabla 6.3: Porcentajes promedios de reducción de mensajes de enrutamiento con el método C-A HC_BN con respecto a 2HC_BN.

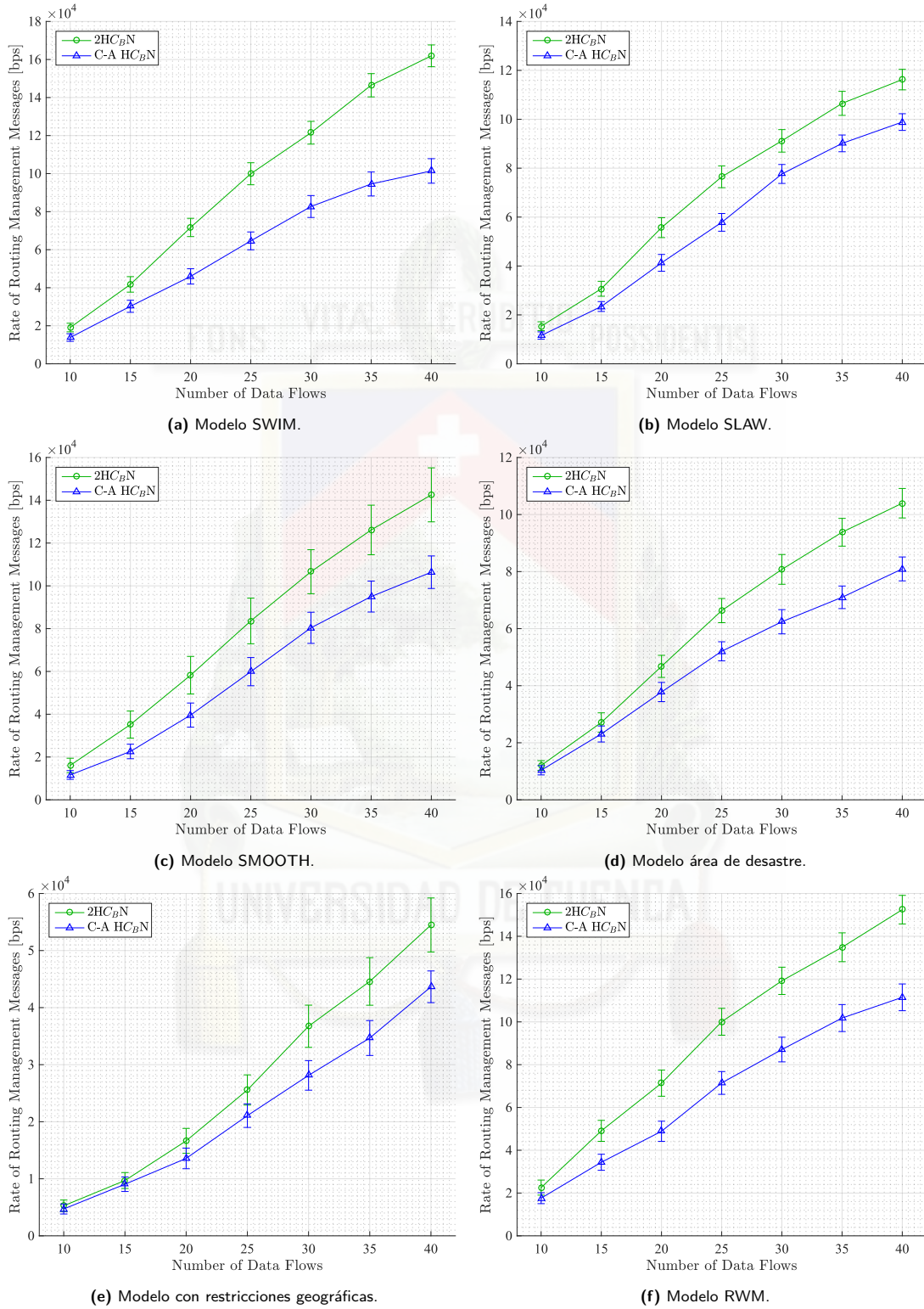


Figura 6.2: Resultados de la tasa de mensajes para gestión de enrutamiento en función del número de flujos para cada modelo de movilidad.

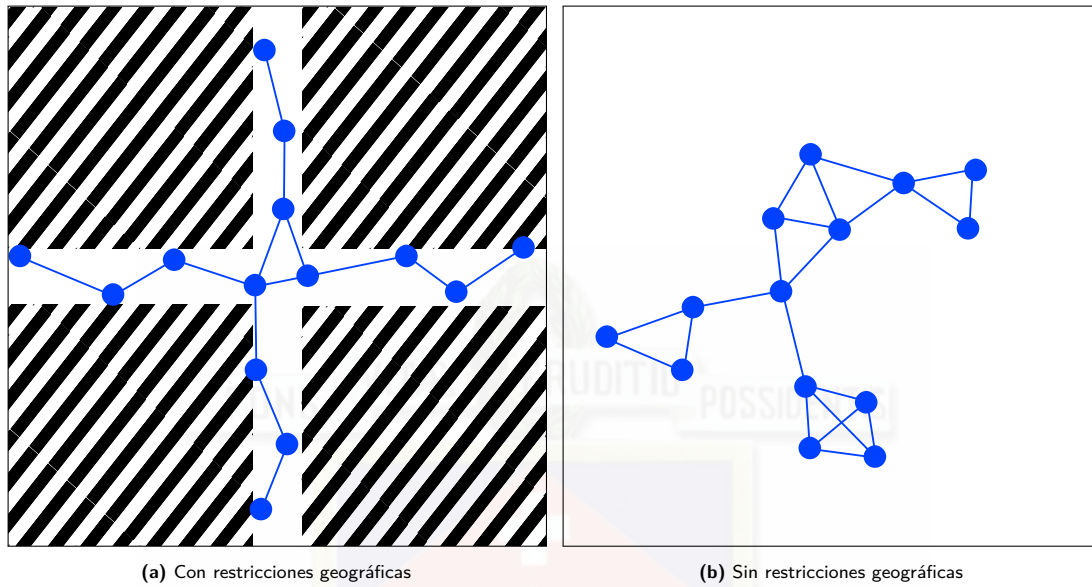


Figura 6.3: Topología generada con 14 nodos en malla

6.1.2. Cantidad de paquetes de datos de unidifusión reenviados en la red

En la Fig. 6.4 se muestra el número total de paquetes de datos de unidifusión reenviados en la red en función de la cantidad de flujos activos para cada modelo de movilidad. En dicha figura se puede observar un crecimiento aproximadamente lineal para cargas de tráfico bajas tanto para $C-A HC_BN$ como para $2HC_BN$. Tal comportamiento lineal se mantiene hasta los 25 flujos en el caso de los modelos SMOOTH, área de desastre y RWM (Fig. 6.4c, 6.4d y 6.4f respectivamente). Mientras que para los modelos SWIM y SLAW (Fig. 6.4a y 6.4b respectivamente), alcanza hasta los 30 flujos activos. Más allá de estas cargas de tráfico, la red empieza a saturarse evidenciando un crecimiento menos significativo. Esto se cumple para la mayoría de modelos de movilidad a excepción del modelo con restricciones geográficas (Fig. 6.4e) en el que la saturación de la red se da con mayores cargas de tráfico. Debido a que este modelo presenta una menor cantidad de mensajes de enrutamiento (Sección 6.1.1), la red permite una mayor cantidad de flujos simultáneos sin saturarse.

En la Tabla 6.4 se muestra el ahorro promedio del total de reenvío de datos del método propuesto $C-A HC_BN$ con respecto a $2HC_BN$ para cada modelo de movilidad. Nuevamente para todos los modelos de movilidad se obtiene una mejora considerable.

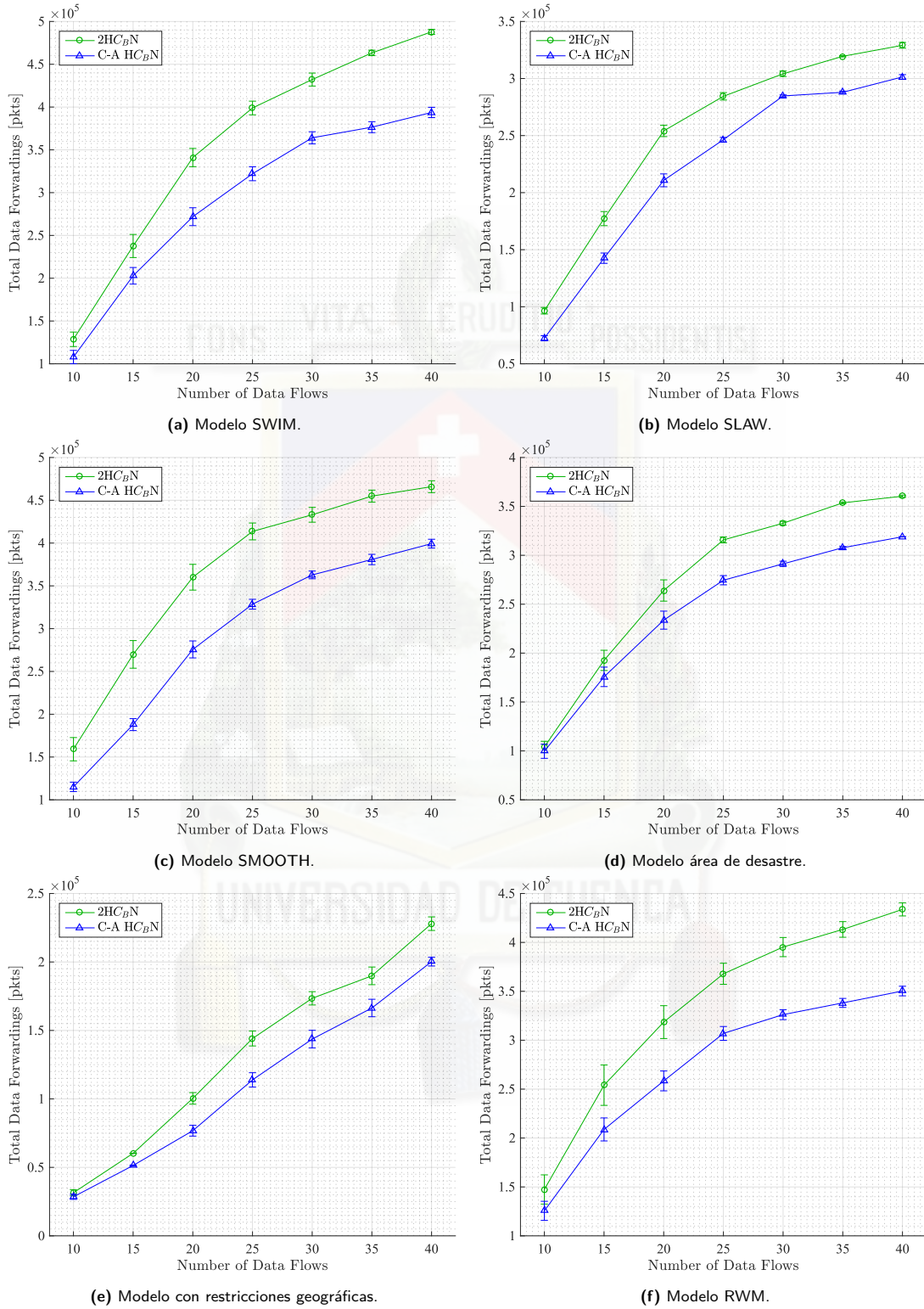


Figura 6.4: Resultados del total de paquetes unidifusión reenviados en función del numero de flujos para cada modelo de movilidad.

SWIM	SLAW	SMOOTH	Disaster Area	Geographic Restrictions	RWM
17.73 %	14.23 %	21.26 %	10.64 %	15.77 %	17.56 %

Tabla 6.4: Porcentaje de reducción del total de paquetes reenviados por la red con el método $C-A HC_BN$ con respecto a $2HC_BN$.

6.1.3. Paquetes recibidos con éxito

Los paquetes recibidos con éxito en función de la carga de tráfico (cantidad de flujos activos) para todos los escenarios, se muestran en la Fig. 6.5. Se aprecia que la red empieza a saturarse con una cantidad significativa de flujos simultáneos (40 flujos). Es por esto que no existe un incremento significativo de paquetes recibidos en ninguno de los dos métodos de control de topología comparados, sobre todo a partir de 40 flujos.

Es evidente una clara mejora para los modelos $SWIM$, $SLAW$, restricciones geográficas y RWM (Fig. 6.5a,6.5b,6.5e,6.5f). Sin embargo, para los modelos $SMOOTH$ y área de desastre (Fig. 6.5c y 6.5d), se observa que no existe una mejora notable en la recepción de paquetes. Las dos curvas muestran valores muy similares. Esto se puede apreciar numéricamente en la Tabla 6.5. Se obtiene un 0.78 % de incremento y 1.43 % de disminución de paquetes recibidos para $SMOOTH$ y área de desastre, respectivamente. Estos valores no son significativos, debido a que se encuentran dentro del intervalo de confianza, revelando que no existe una diferencia considerable.

Con respecto a este parámetro específico, cabe recordar que el modelo de área de desastre, tiene regiones pequeñas con alta conglomeración de nodos como las regiones de compensación de víctimas (CCS). Esto provoca una congestión en ciertas regiones y, por ende, mayor colisión de paquetes.

Por otro lado, en el escenario generado por el modelo de movilidad $SMOOTH$, la cantidad de paquetes entregados exitosamente es similar en ambos mecanismos. Sin embargo, el método propuesto $C-A HC_BN$ reduce el reenvío de paquetes en comparación con $2HC_BN$. Al final, la ventaja de $C-A HC_BN$ con respecto a $2HC_BN$ se verá reflejado en un menor consumo energético como se verá en la Sección 6.1.6.

SWIM	SLAW	SMOOTH	Disaster Area	Geographic Restrictions	RWM
10.63 %	17.25 %	0.78 %	-1.43 %	16.19 %	8.5 %

Tabla 6.5: Porcentaje de incremento de paquetes recibidos con éxito con el método $C-A HC_BN$ con respecto a $2HC_BN$.

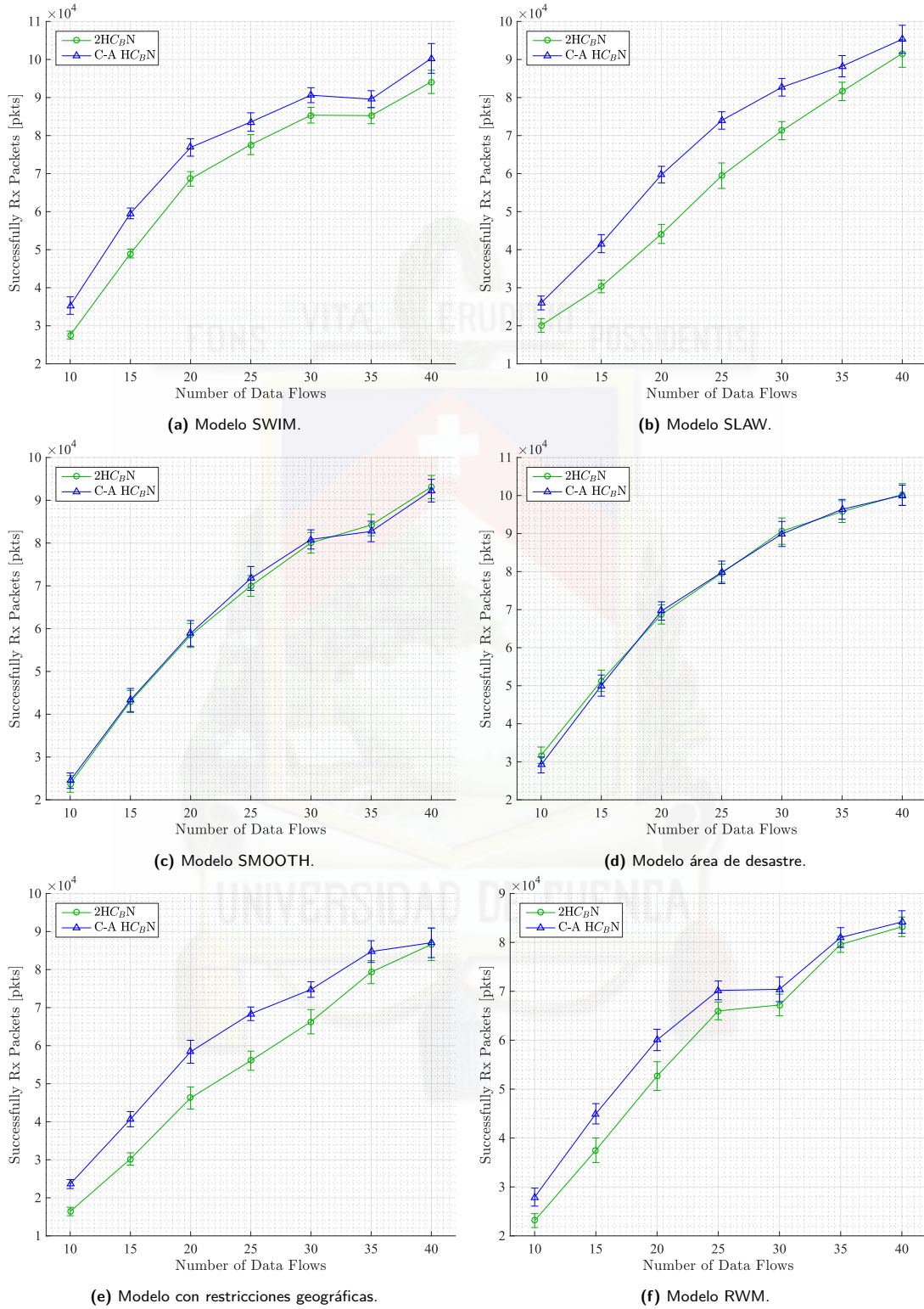


Figura 6.5: Resultados del total de paquetes recibidos con éxito en función de la cantidad de flujos simultáneos para cada modelo de movilidad.

6.1.4. Total de paquetes reenviados por cada paquete recibido con éxito

La relación entre la cantidad de paquetes reenviados y el número de paquetes recibidos con éxito para todos los escenarios, se muestra en la Fig. 6.6. Se observa de forma general que se requiere un menor número de reenvío de paquetes en el método propuesto $C-A HC_BN$ en comparación con el método previo $2HC_BN$. En la mayoría de escenarios se observa que este parámetro tiene un comportamiento creciente debido al incremento de retransmisiones de paquetes a causa del incremento de tráfico. Esto provoca una mayor cantidad de colisiones en toda la red. Para la mayoría de modelos de movilidad este comportamiento creciente se presenta hasta que se alcanza un estado saturado a partir de los 25 o 30 flujos. El modelo $SLAW$, en la Fig. 6.6b, muestra un estado de saturación temprana a partir de 20 flujos. Esto debido a que en este modelo, la cantidad de aristas de la topología reducida es relativamente alta con respecto a los demás modelos de movilidad (Tabla 4.10).

En la Tabla 6.6 se muestra el porcentaje promedio de disminución de datos reenviados por cada paquete recibido con éxito, del método propuesto $C-A HC_BN$ con respecto al método $2HC_BN$. Como se ha tratado en las secciones anteriores, el modelo de área de desastre presenta una disminución relativamente baja en comparación con los demás modelos, debido al particular escenario con regiones de alta densidad de nodos.

SWIM	SLAW	SMOOTH	Disaster Area	Geographic Restrictions	RWM
23.58 %	9.96 %	14.56 %	8.42 %	13.68 %	16.85 %

Tabla 6.6: Porcentaje de disminución de datos reenviados por paquete recibido con éxito con el método $C-A HC_BN$ con respecto a $2HC_BN$.

6.1.5. Eficiencia de la red en términos de proporción de entrega de paquetes

El siguiente parámetro considerado en la presente evaluación, es la eficiencia de la red en términos de proporción de entrega de paquetes (PDR). El PDR se obtiene de la relación de los paquetes recibidos con éxito sobre el total de paquetes transmitidos. Los resultados se muestran en la Fig. 6.7 en la que se evidencia que los modelos de movilidad $SWIM$, $SLAW$, restricciones geográficas y RWM , para todos los flujos de datos, muestran una mejora considerable. Por otra parte, en la Fig. 6.7c correspondiente al modelo $SMOOTH$ y en la Fig. 6.7d correspondiente al área de desastre, la eficiencia no muestra mejoras significativas. En la Tabla 6.7 se confirma que esta mejora es menor al 8 % en estos dos modelos de movilidad. Esto es previsible debido a

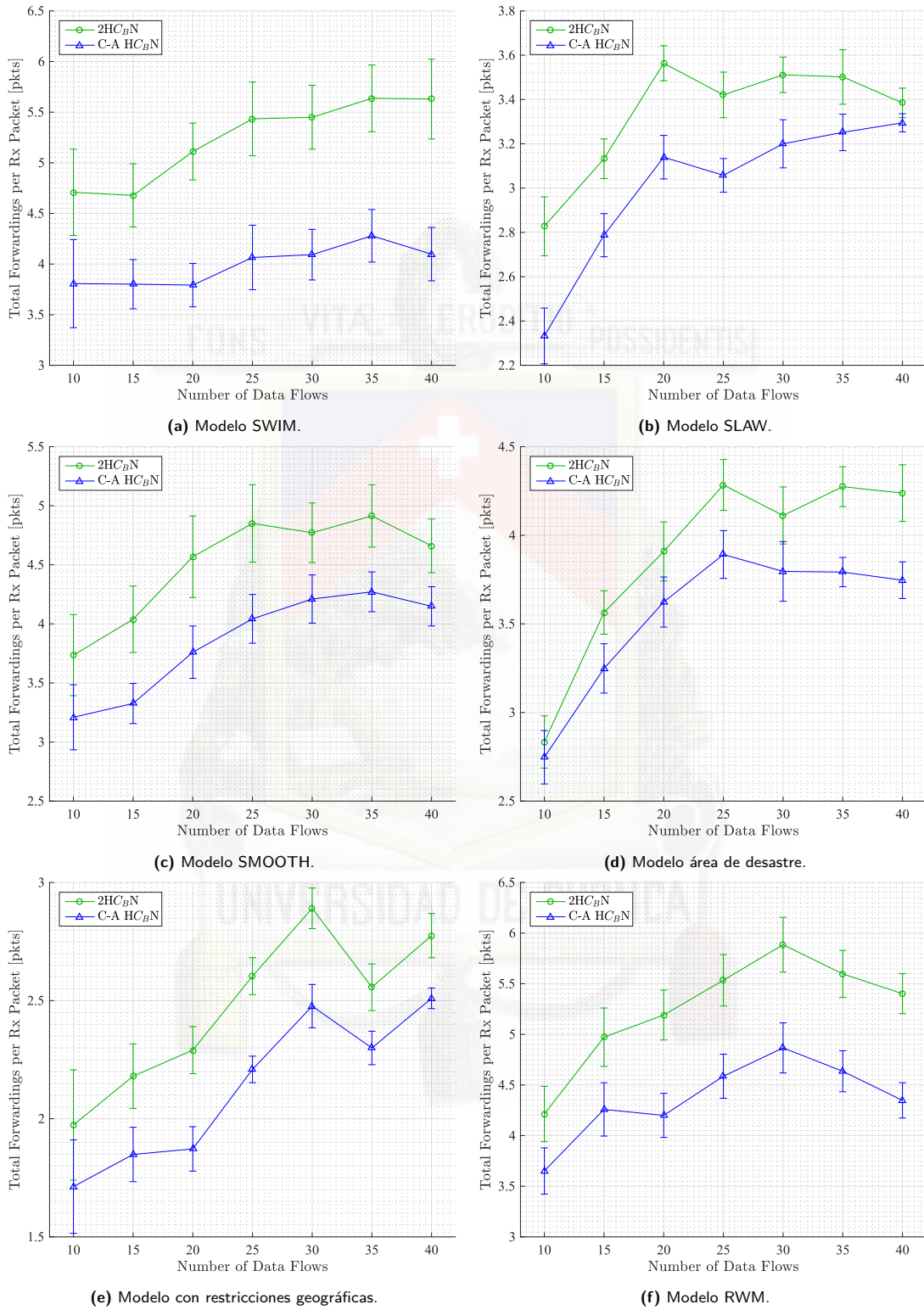


Figura 6.6: Resultados del total de paquetes reenviados por cada paquete recibido con éxito en función de la cantidad de flujos activos para cada modelo de movilidad.

que este parámetro es directamente proporcional a los paquetes recibidos con éxito, parámetro en los cuales tampoco se observaron mejoras significativas (Sección 6.1.3).

Una característica particular que se presenta en todos los modelos de movilidad es la sobrecarga de la red a medida que se incrementa la cantidad de flujos activos en la red. Esto provoca que la eficiencia de la red baje y las curvas de ambos métodos se asemejen para estos puntos. En cambio, al reducir la carga de tráfico lejos de la zona de saturación, se evidencia el incremento de la eficiencia de la red con el mecanismo de control propuesto C-A HC_BN con respecto a 2HC_BN.

SWIM	SLAW	SMOOTH	Disaster Area	Geographic Restrictions	RWM
22.04 %	22.71 %	6.46 %	7.36 %	30.56 %	17.21 %

Tabla 6.7: Porcentaje de incremento de la eficiencia de la red en términos de proporción de entrega de paquetes, del método C-A HC_BN con respecto a 2HC_BN.

6.1.6. Consumo Energético

La mejora en la eficiencia de la red relacionada con la disminución significativa de mensajes de enrutamiento y disminución del reenvío de paquetes, se refleja en la energía total disipada por la red. Para evaluar este parámetro se utiliza el modelo de consumo de energía presentado en [92]. El modelo asume pérdidas debido al canal en el orden de d^2 , donde d es la distancia entre transmisor y receptor. Entonces, para transmitir un paquete de tamaño k , un nodo consume [92]:

$$E_{Tx}(k, d) = E_{elec} \cdot k + \varepsilon_{ampl} \cdot k \cdot d^2 \quad (6.1)$$

y el consumo en la recepción de un paquete es:

$$E_{Rx}(k, d) = E_{elec} \cdot k \quad (6.2)$$

donde, E_{elec} es la energía disipada por los elementos electrónicos del dispositivo y se asume un valor de 50 nJ/bit. Se asume además que el amplificador del transmisor ε_{ampl} disipa 100 pJ/bit/ m^2 .

Para un análisis adecuado se escoge el punto de operación intermedio de 20 flujos activos. En este punto la red no se encuentra sobrecargada, ni tampoco con una carga de tráfico insignificativa. En la Fig. 6.8 se muestra el histograma del consumo energético de los nodos de la red. Se puede observar que una mayor cantidad de nodos tienen un consumo energético más bajo con la propuesta C-A HC_BN. Para tener una mejor apreciación del consumo de energía, en la Tabla 6.8 se resume el consumo energético total promedio de la red para los métodos C-A HC_BN y

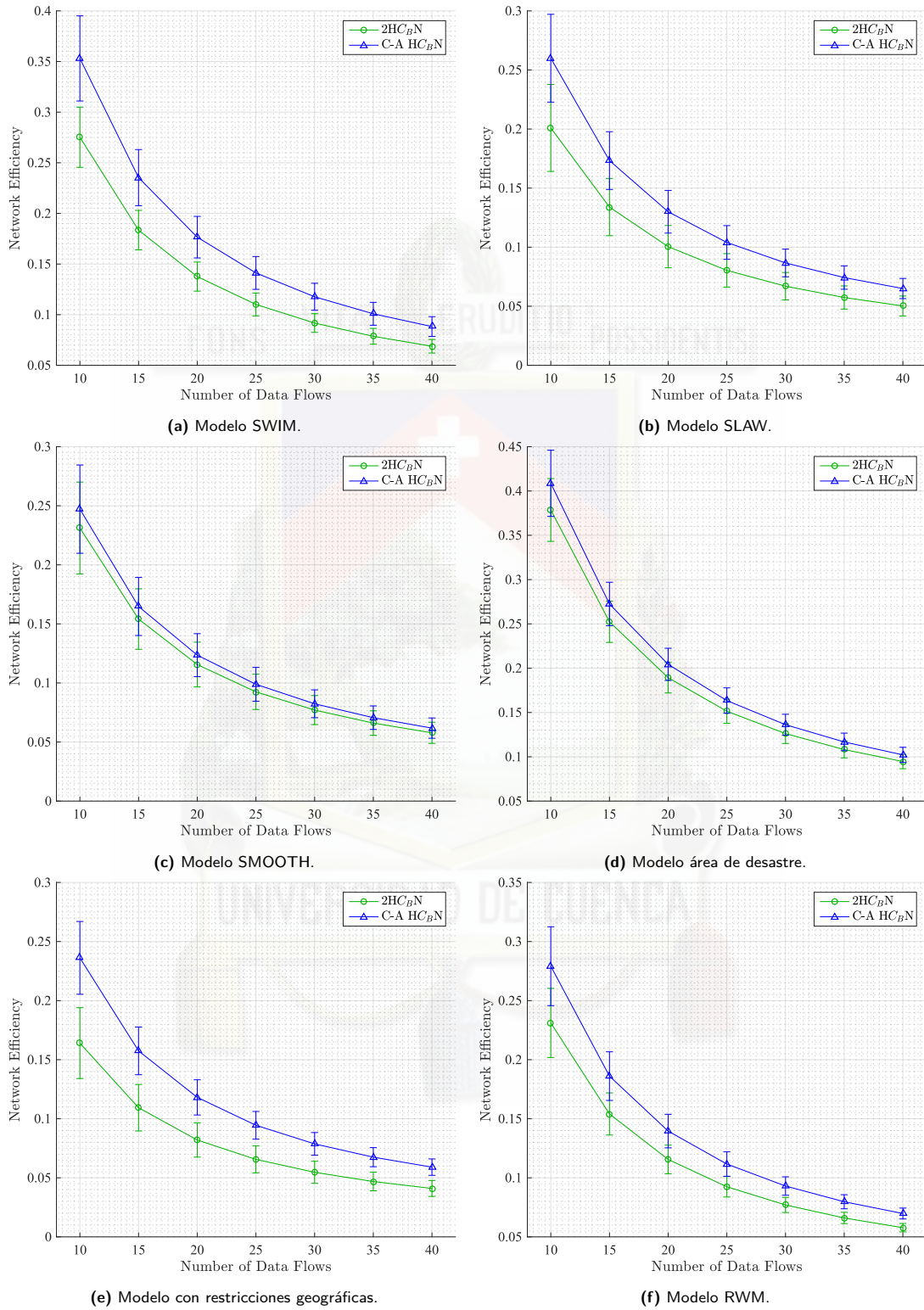


Figura 6.7: Resultados de la eficiencia de la red en términos de la tasa de paquetes entregados en función de la cantidad de flujos activos para cada modelo de movilidad.

$2HC_BN$. La disminución de consumo energético es muy notorio con el método propuesto.

En los modelos de área de desastre y restricciones geográficas se observa el menor consumo de energía en comparación con los demás modelos. Para el modelo de área de desastre, tanto los mensajes de enrutamiento como el número de reenvío de paquetes (Fig. 6.2d y 6.4d respectivamente) tienen una disminución considerable al aplicar el mecanismo de control de topología $C-A HC_BN$ con respecto a $2HC_BN$. Sin embargo, la entrega de paquetes con éxito y la eficiencia de la red (Fig. 6.5d y 6.7d respectivamente) muestran resultados semejantes para ambos mecanismos de control de topología. Al final, la mejora del mecanismo de control propuesto $C-A HC_BN$ se manifiesta en una notable disminución de consumo energético.

Cabe recalcar que para el modelo de restricciones geográficas, debido a la disposición geográfica de los nodos, los mensajes tienden a llegar a su destino con una menor cantidad de saltos y por tanto no se requieren muchas retransmisiones (Fig. 6.6e). Esto se refleja también en un menor consumo energético.

En cambio, el modelo de movilidad SMOOTH (Fig. 6.7c), presenta una eficiencia semejante para los dos mecanismos. Sin embargo, al evaluar el consumo energético, el método $C-A HC_BN$ muestra una disminución considerable, reflejando así una mejora global y significativa en comparación con el método $2HC_BN$.

Model	SWIM	SLAW	SMOOTH	Disaster Area	Geographic Restrictions	RWM
$2HC_BN$	1239.4 J	1035.0 J	1082.6 J	874.7 J	427.1 J	1371.0 J
$C-A HC_BN$	960 J	890.8 J	896.5 J	746.2 J	275.9 J	1028.4 J

Tabla 6.8: Consumo energético promedio de toda la red para 20 flujos activos, con el método $2HC_BN$ y $C-A HC_BN$.

En la Fig. 6.9 se muestra la función de distribución acumulada del consumo de energía. Se observa claramente un mejor comportamiento del mecanismo de control propuesto $C-A HC_BN$. Por ejemplo, en el modelo $SLAW$ (Fig. 6.9b) alrededor del 70 % de nodos consumen menos de 7 Joules para los dos mecanismo de control de topología. Sin embargo, el 30 % de nodos que mayor consumo energético tienen, presentan un ahorro energético con el método $C-A HC_BN$ en comparación a $2HC_BN$.

Para tener una mejor apreciación de este parámetro, en la Tabla 6.9 se resumen los porcentajes de ahorro energético del método $C-A HC_BN$ con respecto a $2HC_BN$.

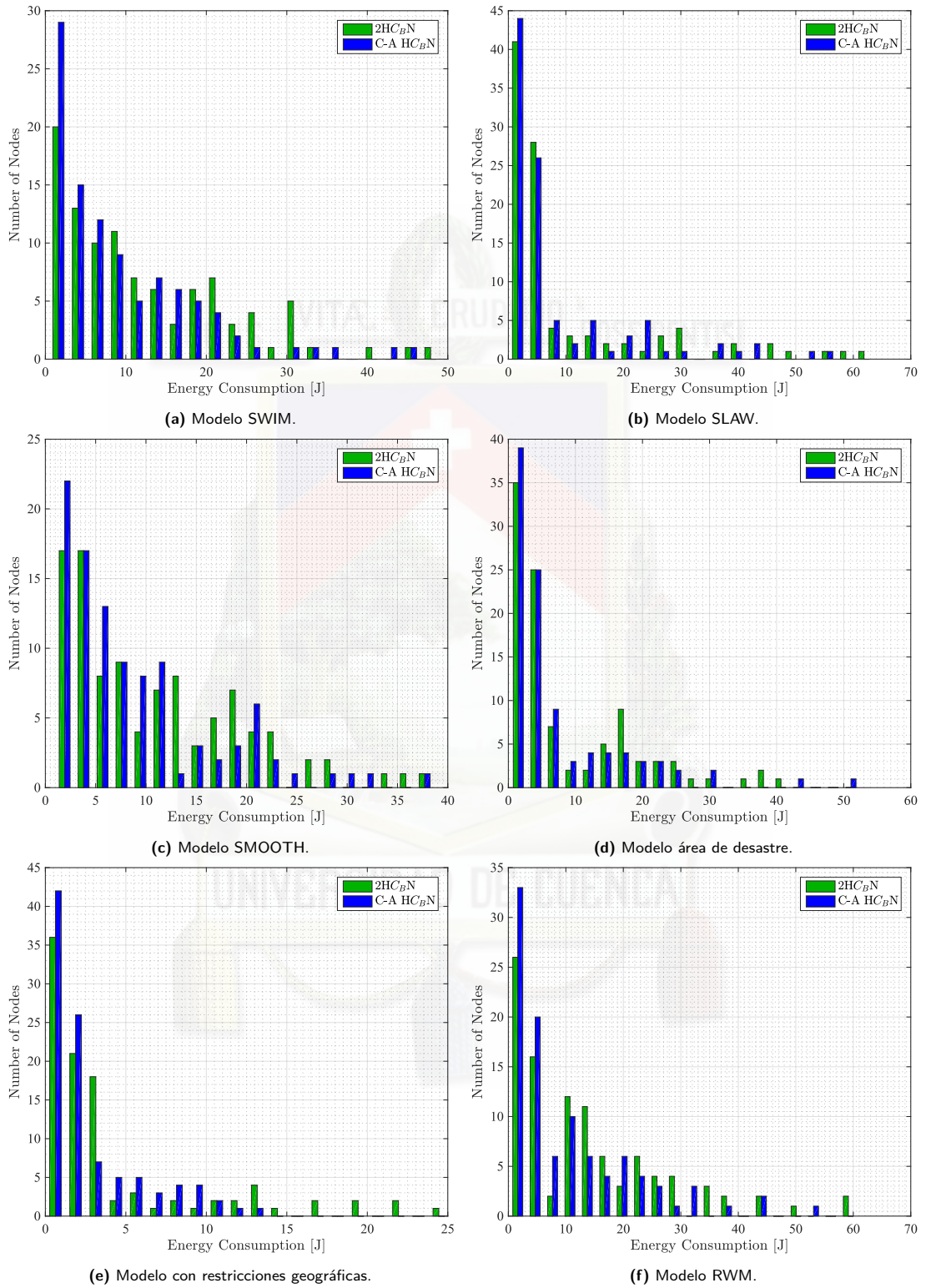


Figura 6.8: Histograma de consumo de energía de los nodos para el caso de 20 flujos activos. Resultados para cada modelo de movilidad.

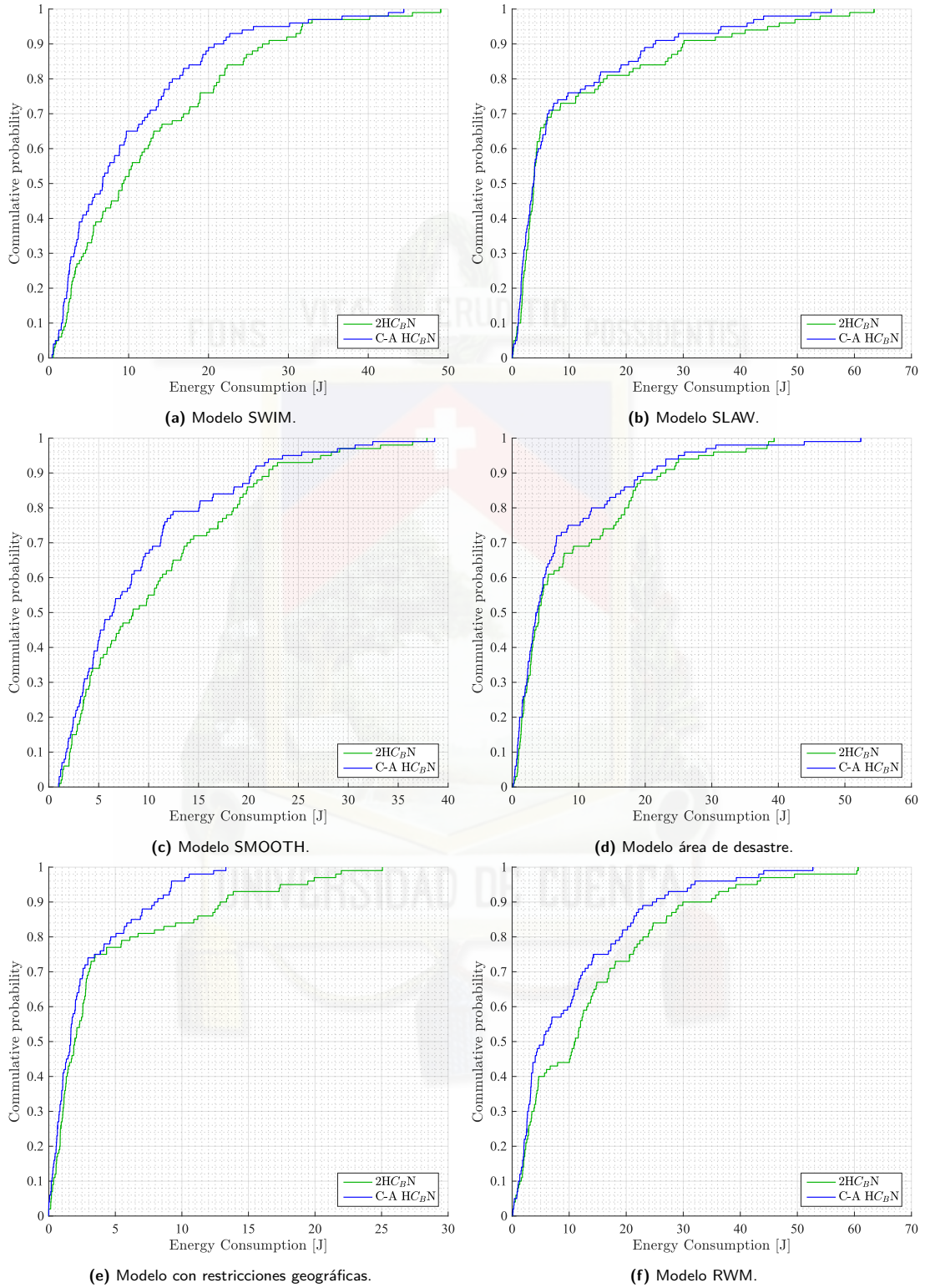


Figura 6.9: Función de distribución acumulada del consumo de energía por nodo para el caso de 20 flujos activos. Resultados para cada modelo de movilidad.

SWIM	SLAW	SMOOTH	Disaster Area	Geographic Restrictions	RWM
22.48 %	13.93 %	17.19 %	14.70 %	35.39 %	24.92 %

Tabla 6.9: Porcentaje de disminución de consumo energético de toda la red para el caso de 20 flujos activos, del método **C-A HC_BN** con respecto al método **2HC_BN**.

6.1.7. Retardo de paquetes extremo-extremo

El último parámetro utilizado para evaluar el impacto de la carga de tráfico es el retardo extremo-extremo. Las Figs. 6.10-6.15 muestran las funciones de distribución acumulada del retardo de paquetes extremo-extremo para cada modelo de movilidad y para las distintas condiciones de carga de tráfico. En todos los casos se aprecia una disminución en el tiempo de retardo. Un comportamiento que se presenta en la mayoría de casos es que, como era de esperarse, a mayor cantidad flujos de datos, se presentan mayores dificultades para tener acceso al canal, y por tanto el retardo aumenta. Por otra parte, la sobrecarga de la red provoca que los mensajes de enrutamiento lleguen tarde y por ende, las selecciones de rutas tardan en resolverse impactando directamente en los retardos extremo-extremo.

Para los modelos: **SLAW** (Fig. 6.11) y restricciones geográficas (Fig. 6.14), se observa que a mayor carga de tráfico, la diferencia de este parámetro de retardo es más significativa. Esto significa que cuando la red se encuentra saturada, el modelo propuesto **C-A HC_BN** reduce en mayor medida la contención, ya que en ambos casos se tiene al mismo tiempo, una mayor cantidad de paquetes recibidos con éxitos (Tabla 6.5).

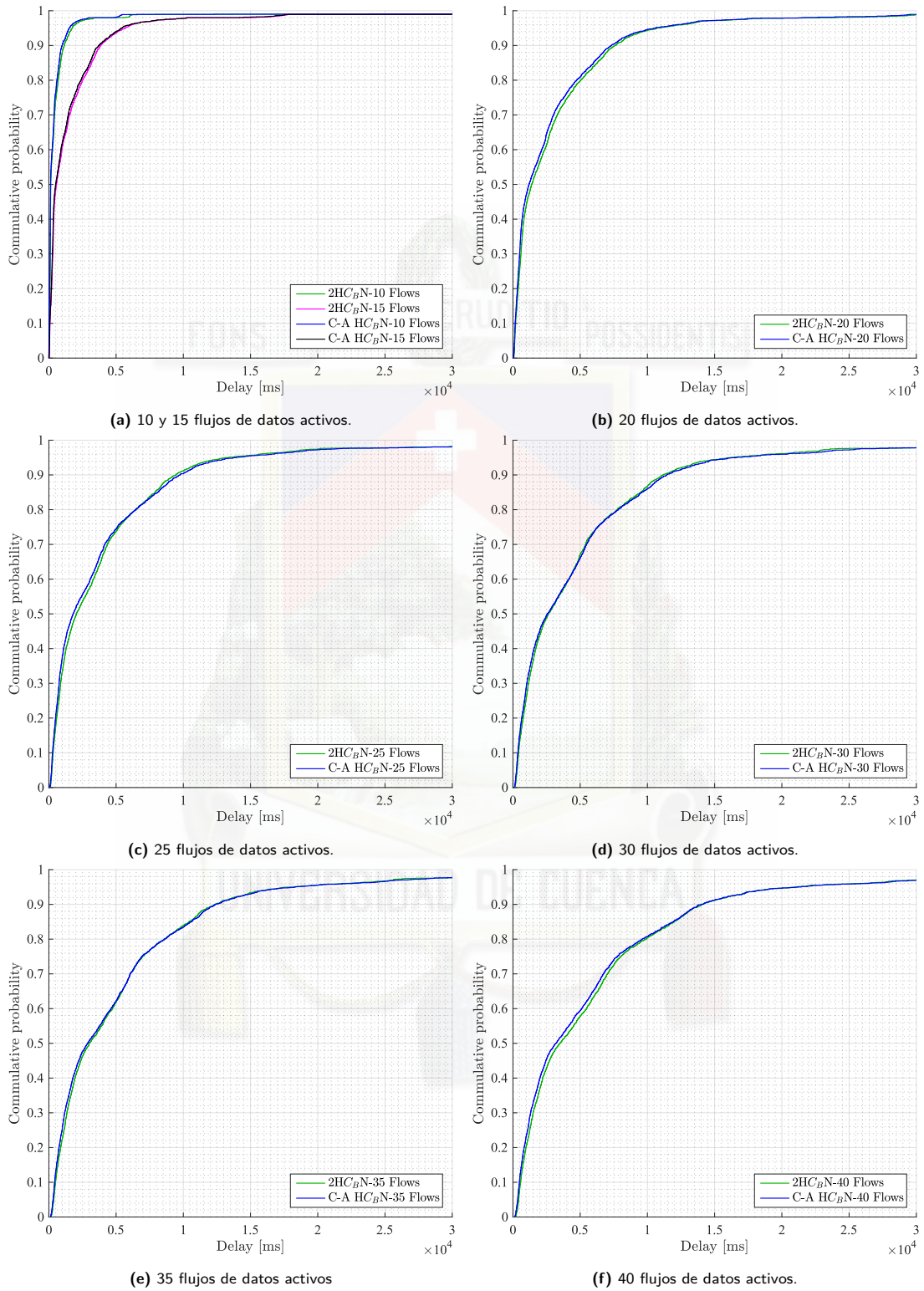


Figura 6.10: Función de distribución acumulada del retardo de paquetes extremo-extremo para el modelo **SWIM**.

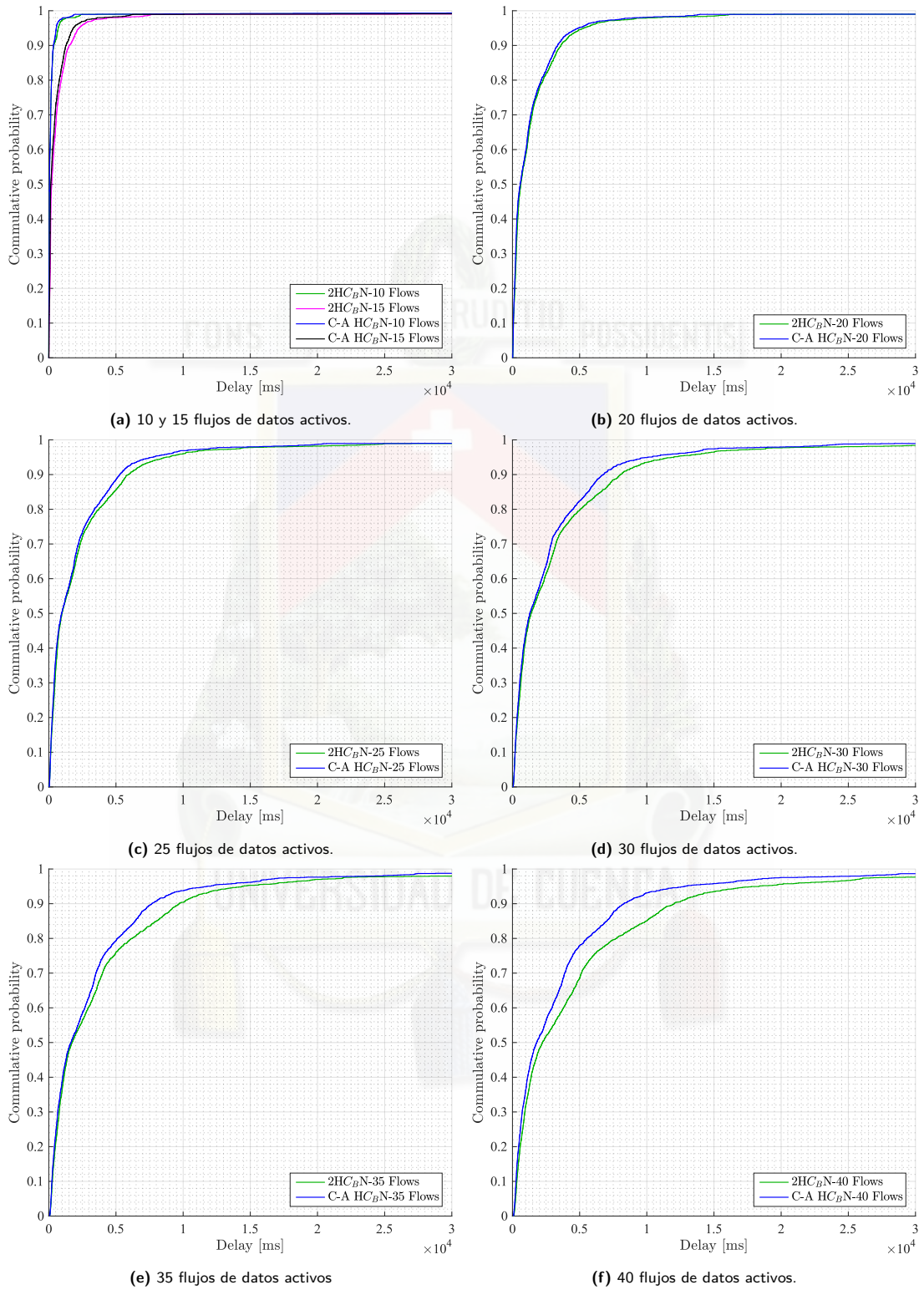


Figura 6.11: Función de distribución acumulada del retardo de paquetes extremo a extremo para el modelo [SLAW](#).

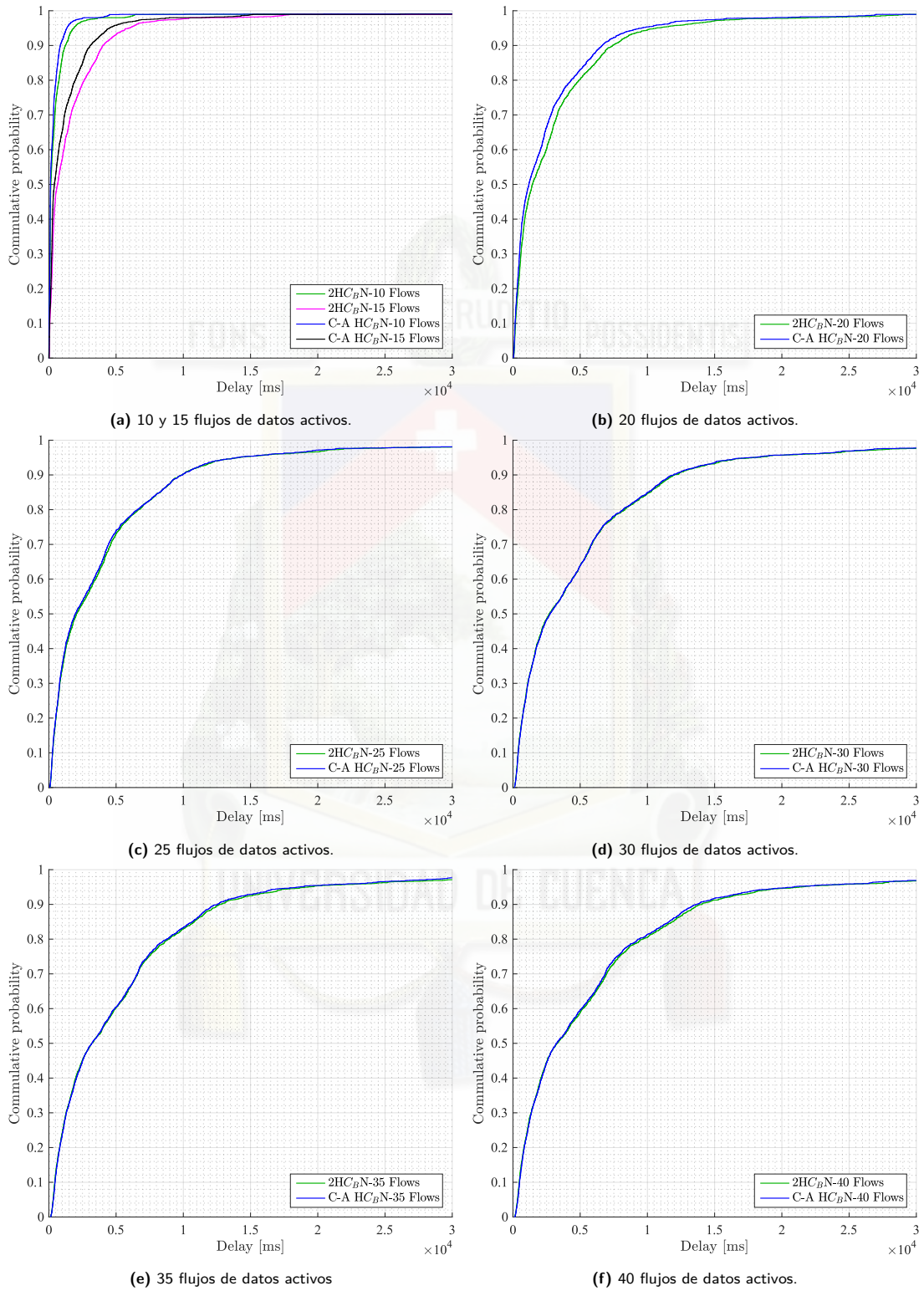


Figura 6.12: Función de distribución acumulada del retardo de paquetes extremo a extremo para el modelo SMOOTH.

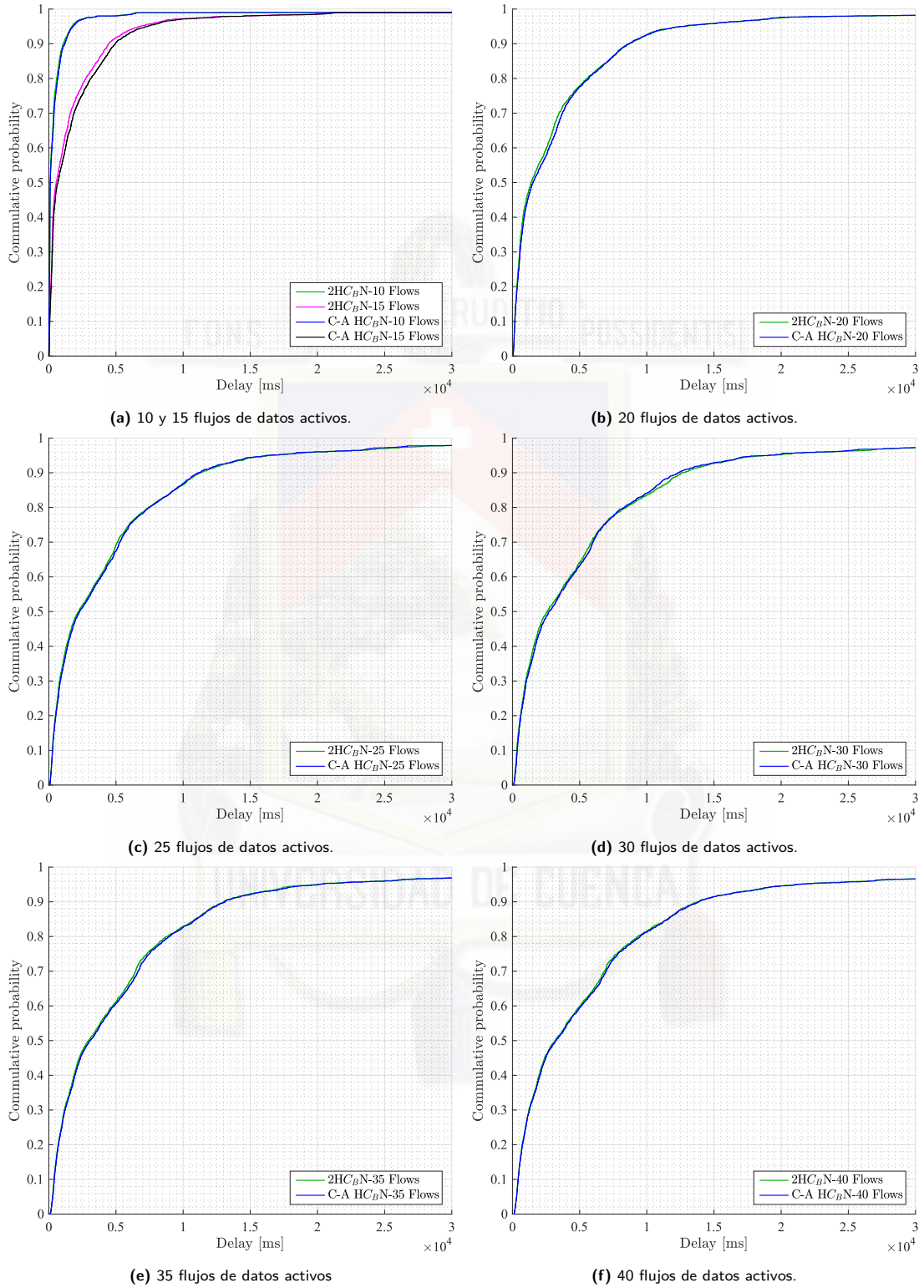


Figura 6.13: Función de distribución acumulada del retardo de paquetes extremo a extremo para el modelo de área de desastre.

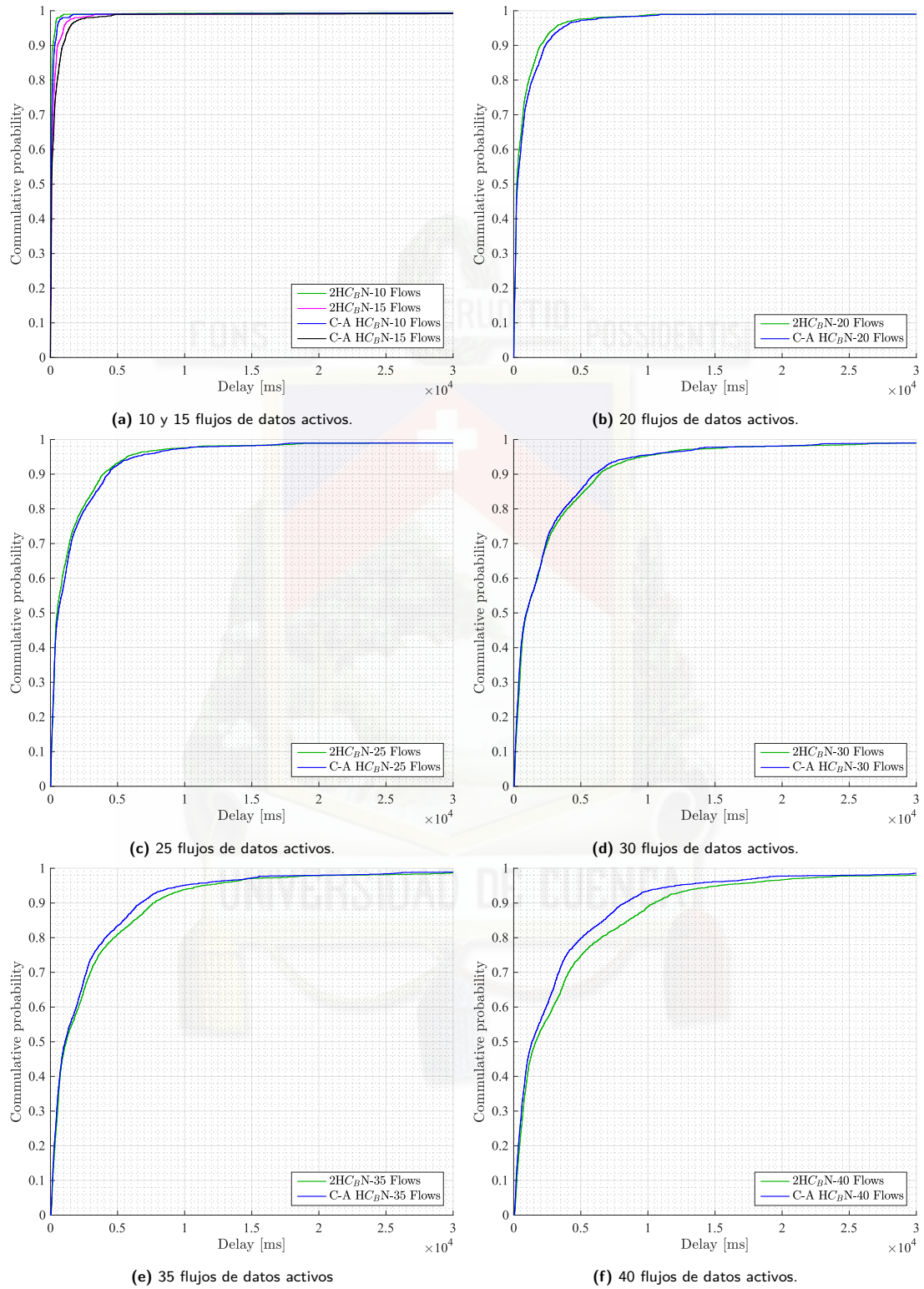


Figura 6.14: Función de distribución acumulada del retardo de paquetes extremo a extremo para el modelo con restricciones geográficas.

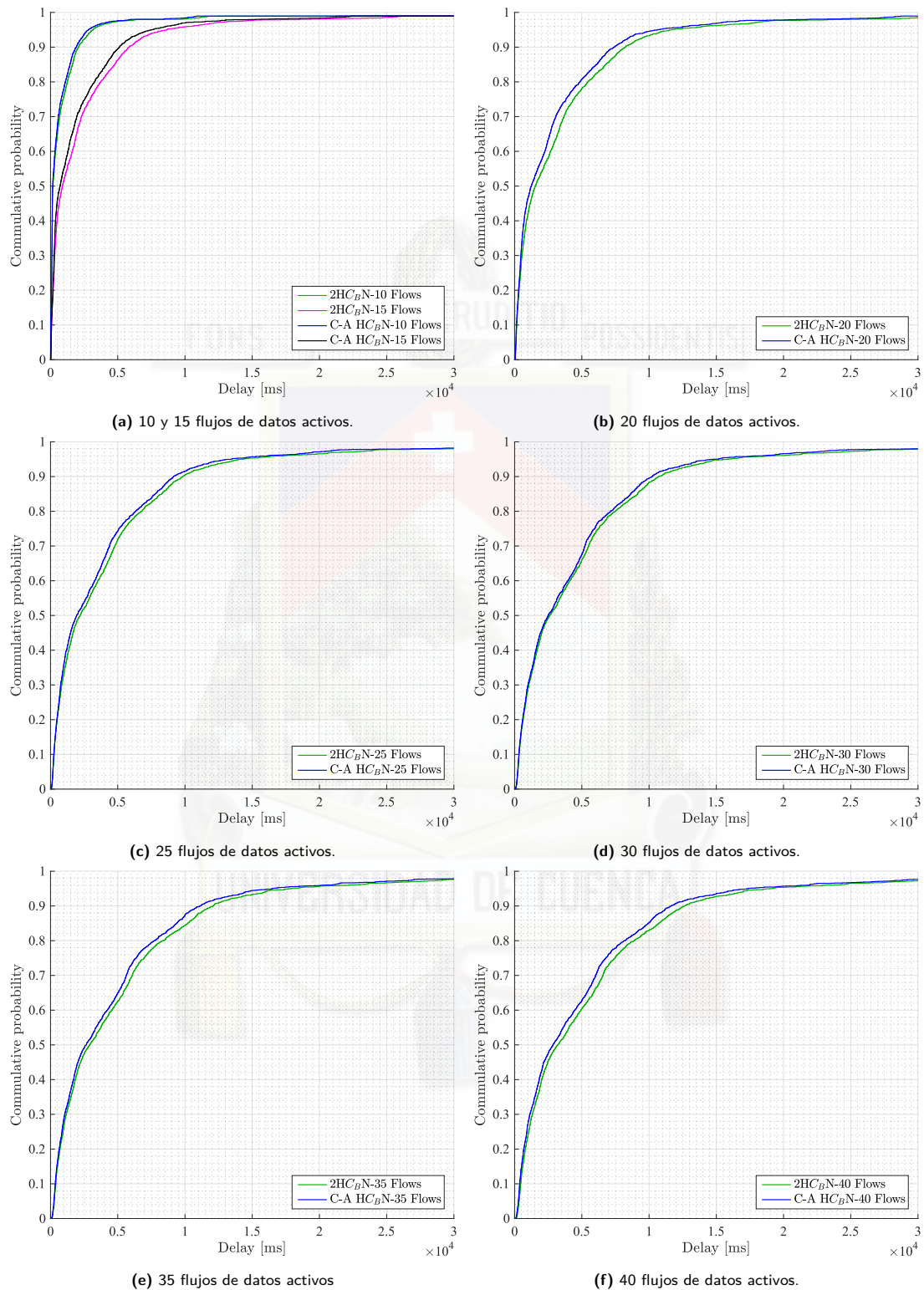


Figura 6.15: Función de distribución acumulada del retardo de paquetes extremo a extremo para el modelo **RWM**.

6.2. Conclusiones

En este capítulo se ha evaluado el desempeño de la red con tráfico **UDP** una vez aplicado el mecanismo de control de topología propuesto **C-A HC_BN**. Para que la evaluación sea confiable, se han llevado a cabo una extensa cantidad de simulaciones con distintas métricas y condiciones de tráfico y para seis modelos de movilidad (5 realistas y un modelo de movimientos aleatorios). En todos los casos se han considerado al menos cuatro capturas instantáneas de la red con 50 ejecuciones estadísticamente independientes para cada uno de los casos. Como era de esperarse, al aplicar cualquiera de los métodos de control de topología, el desempeño de la red es mucho mejor que en el caso de que todos los nodos actúen como enrutadores. Se obtiene una disminución de: mensajes de enrutamiento, reenvío de datos, y consumo energético. Por otro lado la eficiencia de la red aumenta ya que el número de paquetes recibidos exitosamente es mayor. Esto se debe a que las topologías reducidas reducen la cantidad de mensajes en general, la contención, las colisiones de paquetes y la sobrecarga de la red.

Se ha presentado además, la comparación detallada del nuevo método de control de topología propuesto que considera la estructura en comunidades de la red. Comparamos el método (**C-A HC_BN**) con el método previo (**2HC_BN**) que considera únicamente métricas de centralidad. Es posible concluir que, aunque el índice de mensajes recibidos exitosamente no difiere de manera significativa en ambos métodos, el número de mensajes de enrutamiento y retransmisiones sí disminuye significativamente, dando como resultado una red mucho más eficiente y con un menor consumo energético global.





Capítulo 7

Control de topología basado en árboles de expansión

En este capítulo se analiza un método de control de topología que se basa en árboles de expansión y que logra la selección de una menor cantidad de [MR](#).

7.1. Introducción

En el capítulo 6 se evidenció la mejora en el desempeño de una WMN al implementar un mecanismo de control de topología en el cual los nodos pueden conmutar entre los estados MR y MC. Ahora bien, surgen nuevas preguntas como: ¿Se pueden lograr topologías con menor cantidad de MR y conectividad completa de red?, ¿Cómo se puede lograr esto? y ¿Cuál será el desempeño de tales redes con tráfico en tiempo real en comparación con la propuesta C-A HC_BN?

La respuesta a la primera pregunta es afirmativa. Al utilizar el método de control de topología C-A HC_BN se puede observar fácilmente que existen algunos nodos MR que podrían resultar redundantes en los diversos caminos entre distintos puntos. Por tanto, al omitir estos nodos redundantes será posible obtener una topología aún más reducida.

En cuanto a la segunda pregunta, en este trabajo se evalúa específicamente un mecanismo de control de topología basado en árboles de expansión. En este capítulo se describen las topologías obtenidas con este método y se comparan los resultados con el mecanismo C-A HC_BN propuesto. El objetivo es por tanto, contestar la tercera pregunta y evaluar si efectivamente se obtiene un mejor desempeño general de la red.

7.2. Control de topología basado en Método de Árboles de Expansión

El método ST (Spanning Tree) es una alternativa de control de topología que no considera implícitamente la estructura en comunidades de la red. Consiste en escoger a los nodos enrutadores en base a una reducción topológica de la red basada en un algoritmo de árbol de expansión mínimo. El problema del árbol de expansión mínimo fue establecido y resuelto por primera vez por Otakar Borůvka en 1926 [93–95]. Después de eso, varios algoritmos para hallar el árbol de expansión mínimo han sido propuestos. Varios de éstos se describen en [96–99]. El problema es el siguiente: Dado un grafo conectado con pesos en sus aristas, encontrar el árbol de expansión (mínimo conjunto de aristas que conecta todos los vértices) del grafo con mínimo peso.

El algoritmo que se ha utilizado para este trabajo calcula el backbone de un grafo utilizando un algoritmo basado en árboles de expansión mínimo. El núcleo de red resultante es aquel árbol de expansión mínimo que une todos los nodos de la red de tal manera que la suma de los pesos de las aristas sea la máxima. Este algoritmo fue desarrollado por los autores de [100, 101], disponible en código Matlab en línea [102].

En la Fig. 7.1b se observa el árbol de expansión mínimo generado para el modelo de movilidad SWIM en el segundo $t = 30$ cuya topología completa se ilustra en la Fig. 7.1a.

En este punto, el desafío es entonces elegir a los nodos enrutadores que conformen el núcleo de la red resultante (*backbone*). La propuesta natural en este caso es escoger como enrutadores aquellos nodos que no sean nodos terminales (*leaf nodes*) en el árbol de expansión mínimo, es decir, los nodos con más de un enlace. Así pues, la topología reducida con este método es aquella que se muestra en la Fig. 7.1c en la que los nodos de color azul son aquellos escogidos como enrutadores y los nodos de color verde son nodos cliente. El *backbone* correspondiente se muestra en la Fig. 7.1d. Nótese que la red siempre tiene fragmentación cero, es decir que siempre existe conectividad completa en la red (excepto por aquellos nodos que se encuentren originalmente aislados, en caso de haberlos).



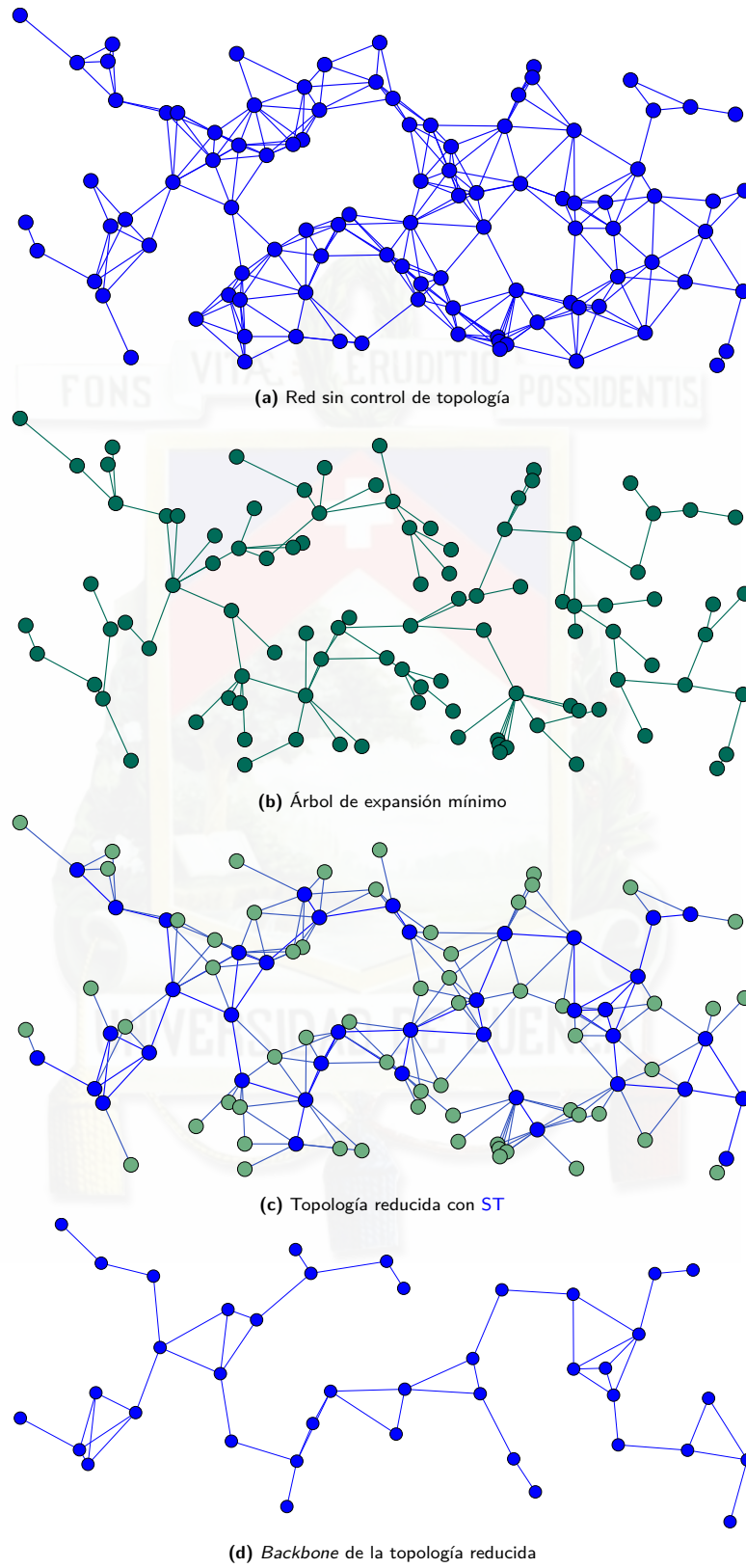


Figura 7.1: Topologías creadas con el método de árbol de expansión (ST) para el modelo SWIM en $t = 30\text{seg}$

7.3. Análisis de las topologías construidas

Al igual que en la Sección 4.4, aquí se analizan las topologías reducidas resultantes al aplicar el método ST para la selección de MR. Recordar que se consideran 6 modelos de movilidad distintos y se ejecutan simulaciones durante 500 segundos, con una actualización de topología cada 5 segundos.

En la Tabla 7.1 se muestra la cantidad promedio de nodos MR que se seleccionan para el caso de C-A HCBN y para ST para los seis modelos de movilidad. Los valores para el caso de C-A HCBN son obtenidos de la Tabla 4.8. La disminución de nodos seleccionados es notoria. Existe una diferencia media comprendida entre 4 % y 11 % de nodos seleccionados para todos los modelos de movilidad.

Model	C-A HCBN (promedio)	ST (promedio)
SWIM	40.28	35.06
SLAW	36.34	25.48
SMOOTH	41.43	37.16
Disaster A.	36.33	30.92
Geographic R	37.66	28.07
RWM	42.19	36.33

Tabla 7.1: Promedio de enrutadores seleccionados para los métodos C-A HCBN vs ST en escenarios de 100 nodos y 500 segundos de duración

Por otro lado es necesario evaluar la cantidad de veces que un nodo conmuta de estado, entre MC y MR, durante los 500 segundos de simulación. En la Tabla 7.2 se muestra el valor medio de este parámetro para los dos casos. Los valores para C-A HCBN son obtenidos de la Tabla 4.9. Por su parte, los histogramas completos de cambios de estado para ambos métodos se pueden apreciar en la Fig. 7.2. Los resultados correspondientes al método C-A HCBN se muestran con azul, mientras que los resultados de ST están de color magenta. A simple vista se puede apreciar que la mayor cantidad de nodos en el método ST tienen pocos cambios de estado comparado con C-A HCBN. Esto puede ser una ventaja desde el punto de vista de la estabilidad de la red, sin embargo, podría representar un consumo de energía excesivo de estos nodos.

Para el caso de SWIM y Área de desastre que presentan un mayor grado de movilidad debido a la considerable variación y reconfiguración de grupo de los nodos, se tiene una media de 11.11 y 12.36 cambios de estado, lo que equivale a decir que un nodo en promedio, tiene una tasa de cambio de estado de 1.3332 veces/minuto y 1.4832 veces/minuto respectivamente. Estos valores corresponden aproximadamente a la mitad de la tasa de cambios que en el caso del método C-A HCBN. Para los demás escenarios los que la movilidad de nodos es baja, se tienen valores de media entre 4.52 y 5.89 los que corresponden a tasas de cambios de estado de 0.5424 a 0.7068

veces/minuto. Estos valores corresponden a $1/3$ aproximadamente de los valores obtenidos con el método C-A HC_BN .

Model	C-A HC_BN (promedio)	ST (promedio)
SWIM	23.36	11.11
SLAW	15.96	5.52
SMOOTH	12.2	4.52
Disaster A.	22.77	12.36
Geographic R	13.35	5.73
RWM	15.78	5.89

Tabla 7.2: Promedio de cambios de estado de los nodos para los métodos C-A HC_BN vs ST en escenarios de 100 nodos, 500 segundos de duración, y diferentes modelos de movilidad

Finalmente en la Fig. 7.3 podemos apreciar la evolución del número de las aristas en la red durante los 500 segundos de cada escenario. Los resultados correspondientes al método C-A HC_BN se muestran con azul, mientras que los resultados de ST están de color magenta. Como el número de nodos MR seleccionados en el método ST es menor con respecto a C-A HC_BN , el número de aristas de las topologías reducidas también decrece en la mayor parte del tiempo.

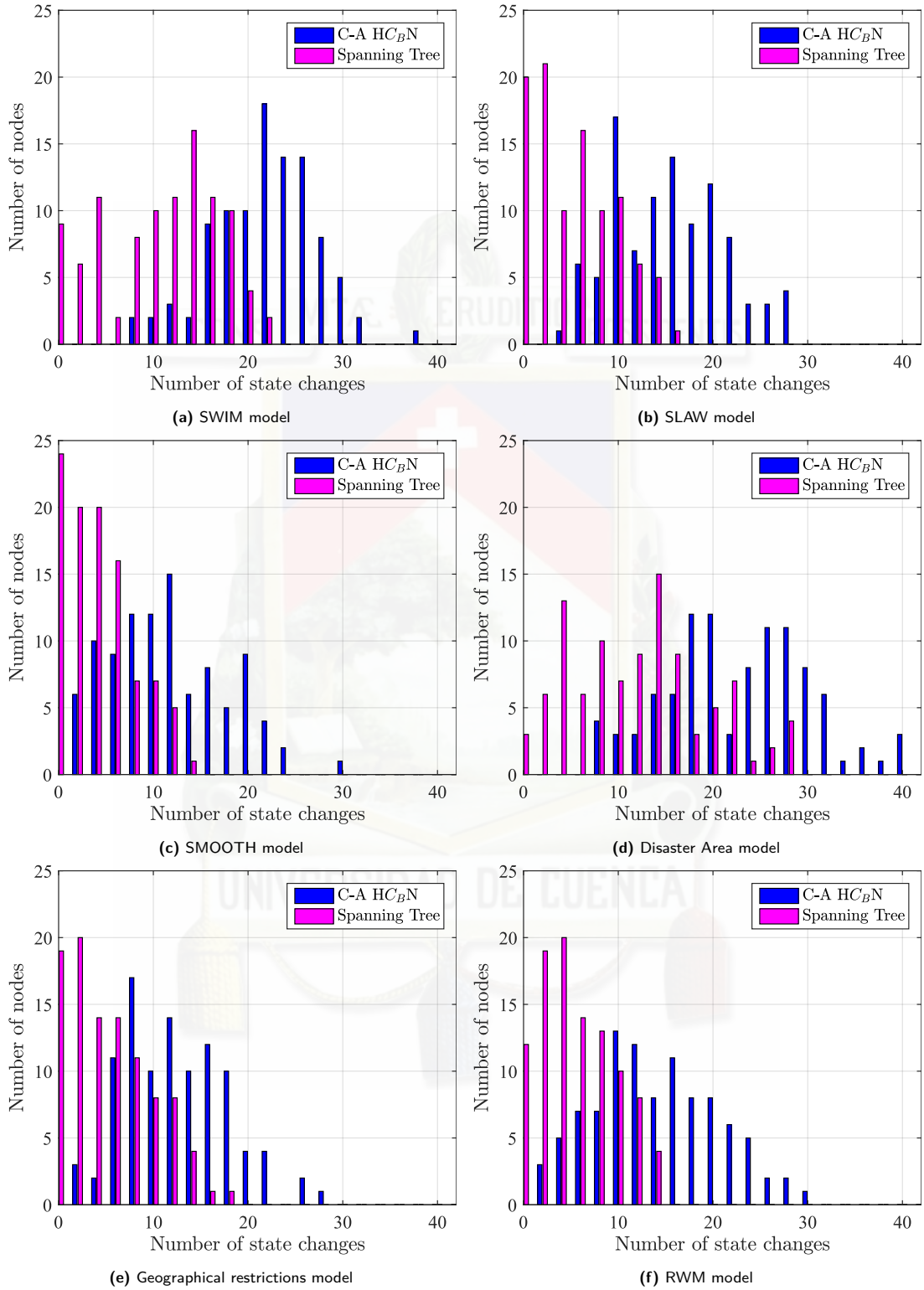


Figura 7.2: Histogramas de la cantidad de cambios de estado de los nodos, para los métodos C-A HC_BN y ST para el tiempo total de simulación.

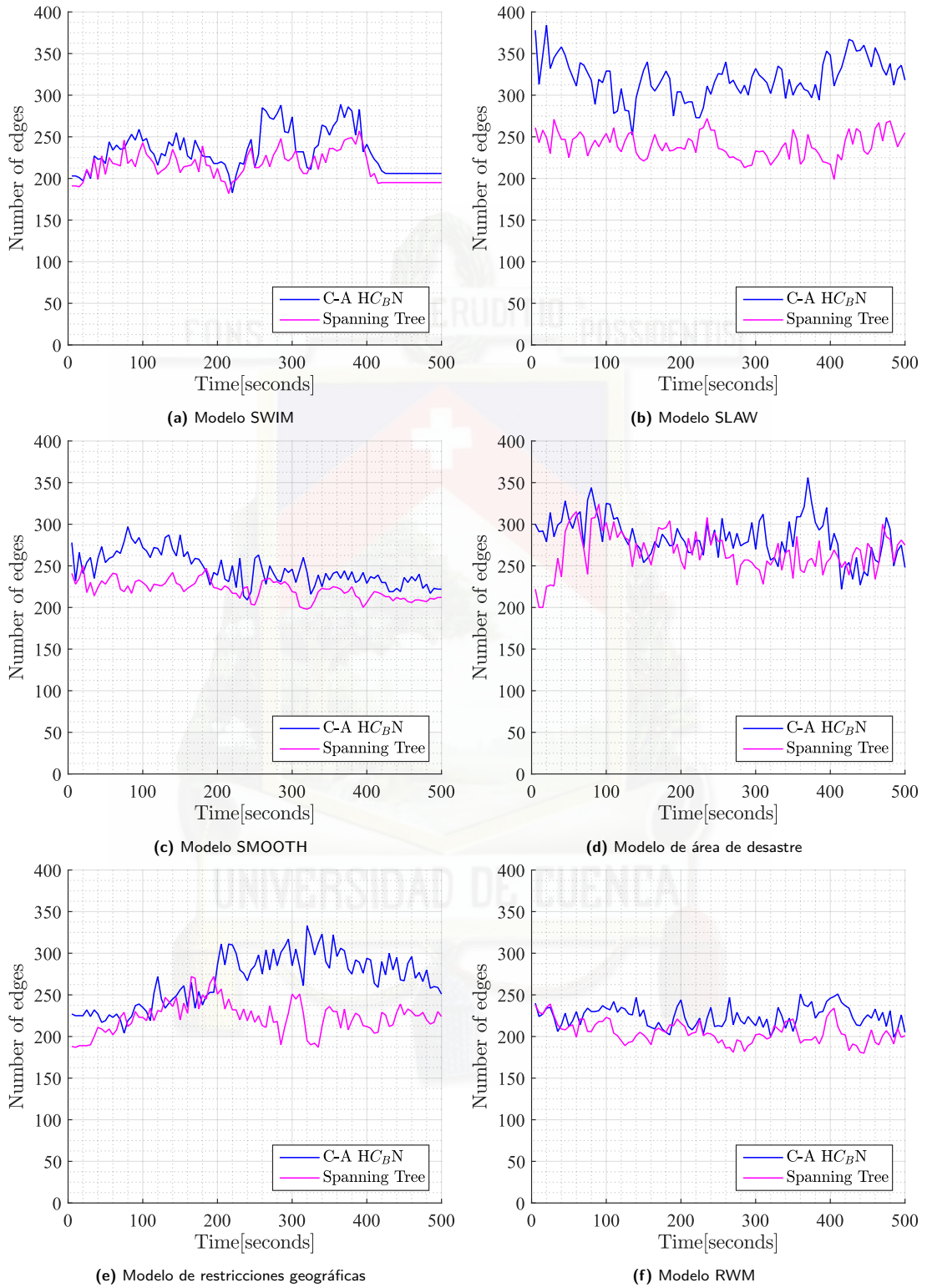


Figura 7.3: Evolución en el tiempo del número de aristas en la red a través de los 500 segundos de duración de cada escenario, para los métodos C-A HC_BN y ST

7.4. Impacto de la carga de tráfico en la red

En esta sección se evalúa el impacto de la carga de tráfico utilizando el mecanismo de control basado en la selección de enrutadores mediante el método **ST** basado en árboles de expansión. Este método selecciona en general una menor cantidad de enrutadores en comparación con **C-A HC_BN** como se indica en la Tabla 7.1.

Los resultados obtenidos de forma gráfica se ilustran en las Figs. 7.4-7.10. De color azul se muestran los resultados correspondientes al método **C-A HC_BN** y de color magenta aquellos correspondientes al método **ST**. Un resumen de los resultados obtenidos se muestra en la Tabla 7.3. En la columna (ST)–(C-A HC_BN) se muestra la diferencia que existe entre, el porcentaje de mejora del método **ST** con respecto al caso con todos los nodos como enrutadores (*All Routers*), y el método **C-A HC_BN** con respecto a *All Routers*.

Para la tasa de mensajes de enrutamiento se observa que el mecanismo de control basado en árboles de expansión disminuye en un 2.88 % en promedio para todos los modelos de movilidad en comparación con el método propuesto **C-A HC_BN**. Este ahorro es esperado debido a la menor selección de nodos **MR**, por lo que se resuelve la selección de rutas con una menor cantidad de mensajes de enrutamiento. De igual manera, la menor cantidad de rutas provoca que el total de datos reenviados por la red ahorre un 6.19 % promedio de paquetes con el mecanismo **ST** en comparación con **C-A HC_BN**.

En la recepción de paquetes con éxito se observa que el mecanismo **ST** tiene un detrimento del 2.75 % promedio con respecto a **C-A HC_BN**. Debido a que el backbone contiene muy pocas rutas posibles, se produce una reducción de trayectorias posibles entre los nodos. Esta disminución de nodos **MR** y rutas, provoca un alto tráfico en los enrutadores escogidos. Este tráfico se traduce en una congestión de paquetes, lo que finalmente provoca una disminución en la recepción exitosa de datos. Por ende, la eficiencia de la red (en términos de **PDR**), en el mecanismo propuesto **C-A HC_BN** tiene una superioridad del 2.94 % promedio con respecto al método **ST**.

Finalmente, el consumo de energía en el modelo basado en árboles de expansión presenta un ahorro del 2.47 % promedio con respecto al método propuesto **C-A HC_BN**. Esto se debe a la menor cantidad de mensajes reenviados por cada mensaje recibido con éxito, donde también se observa un ahorro del 3.93 % promedio.

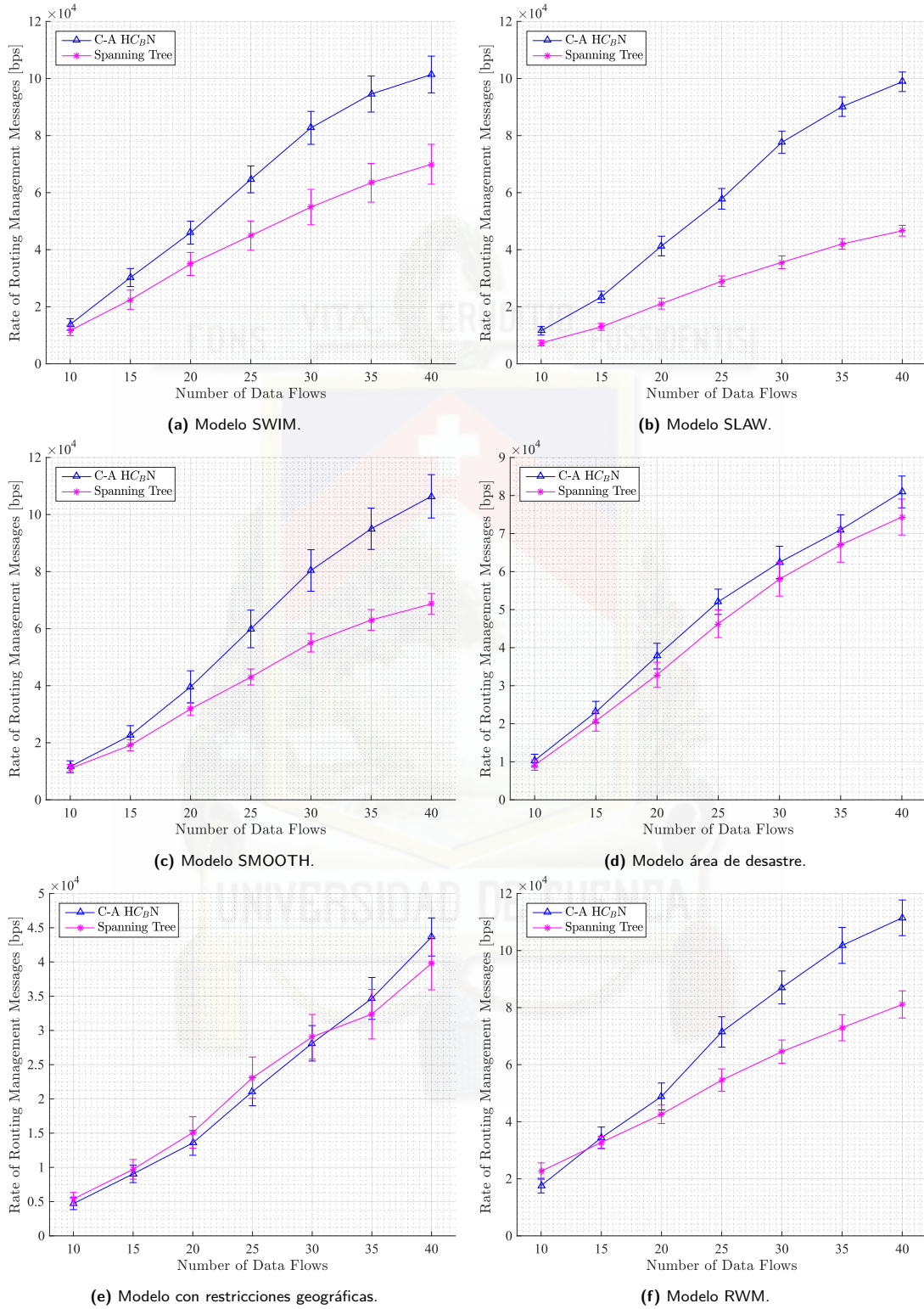


Figura 7.4: Resultados de la tasa de mensajes para gestión de enrutamiento en función del número de flujo para cada modelo de movilidad.

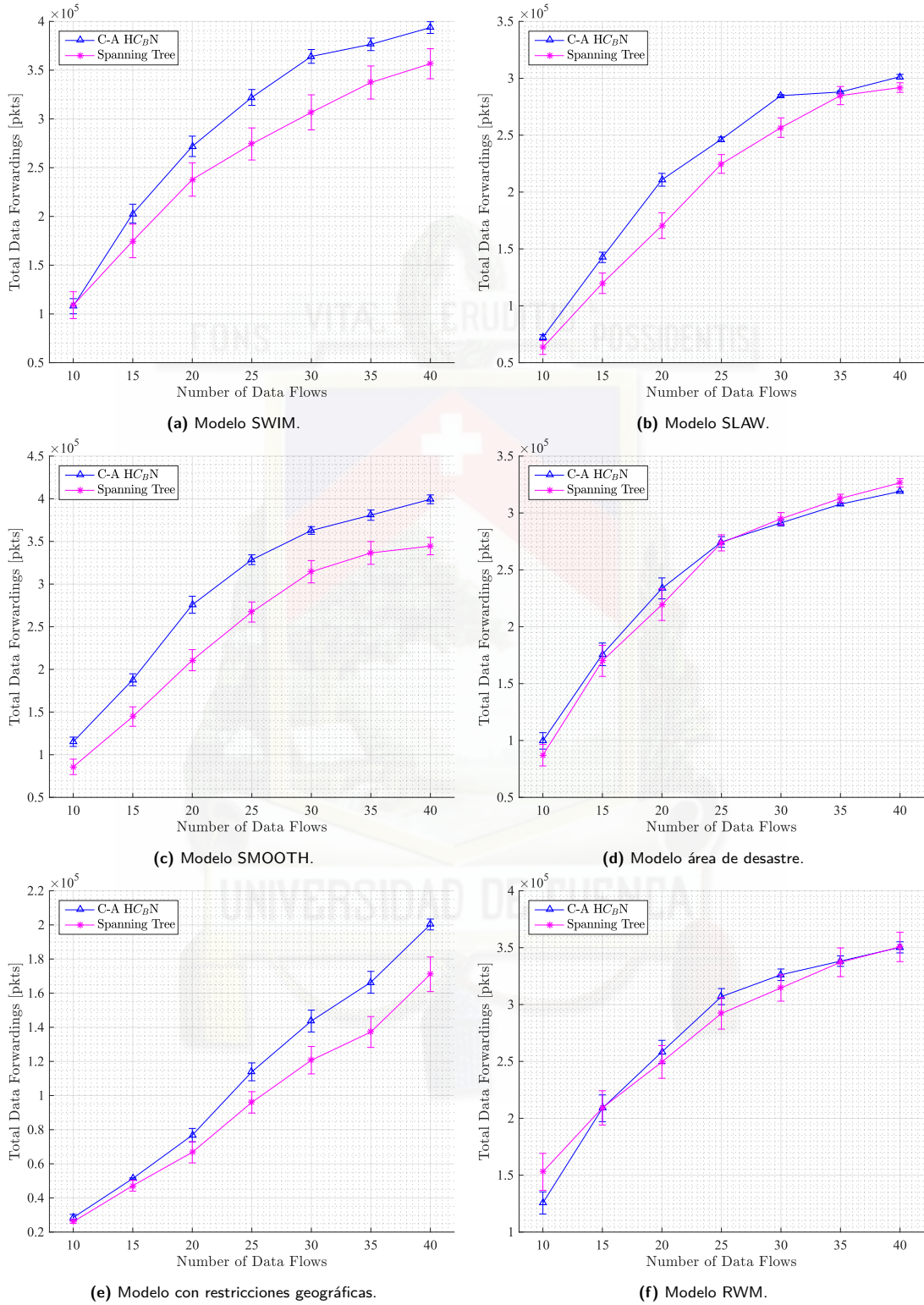


Figura 7.5: Resultados del total de datos reenviados en función del numero de flujo para cada modelo de movilidad.

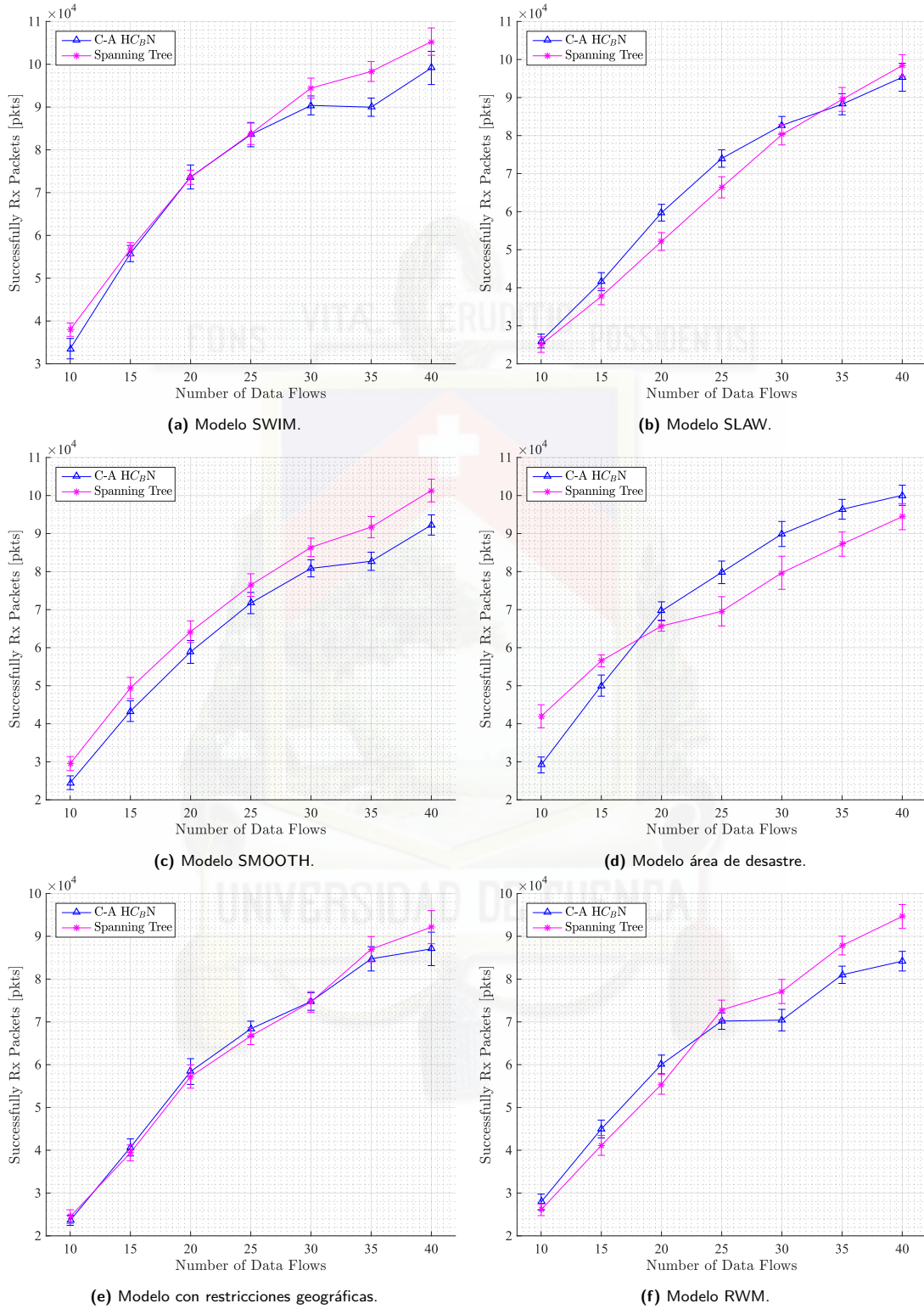


Figura 7.6: Resultados del total de paquetes recibidos con éxito en función del número de flujo para cada modelo de movilidad.

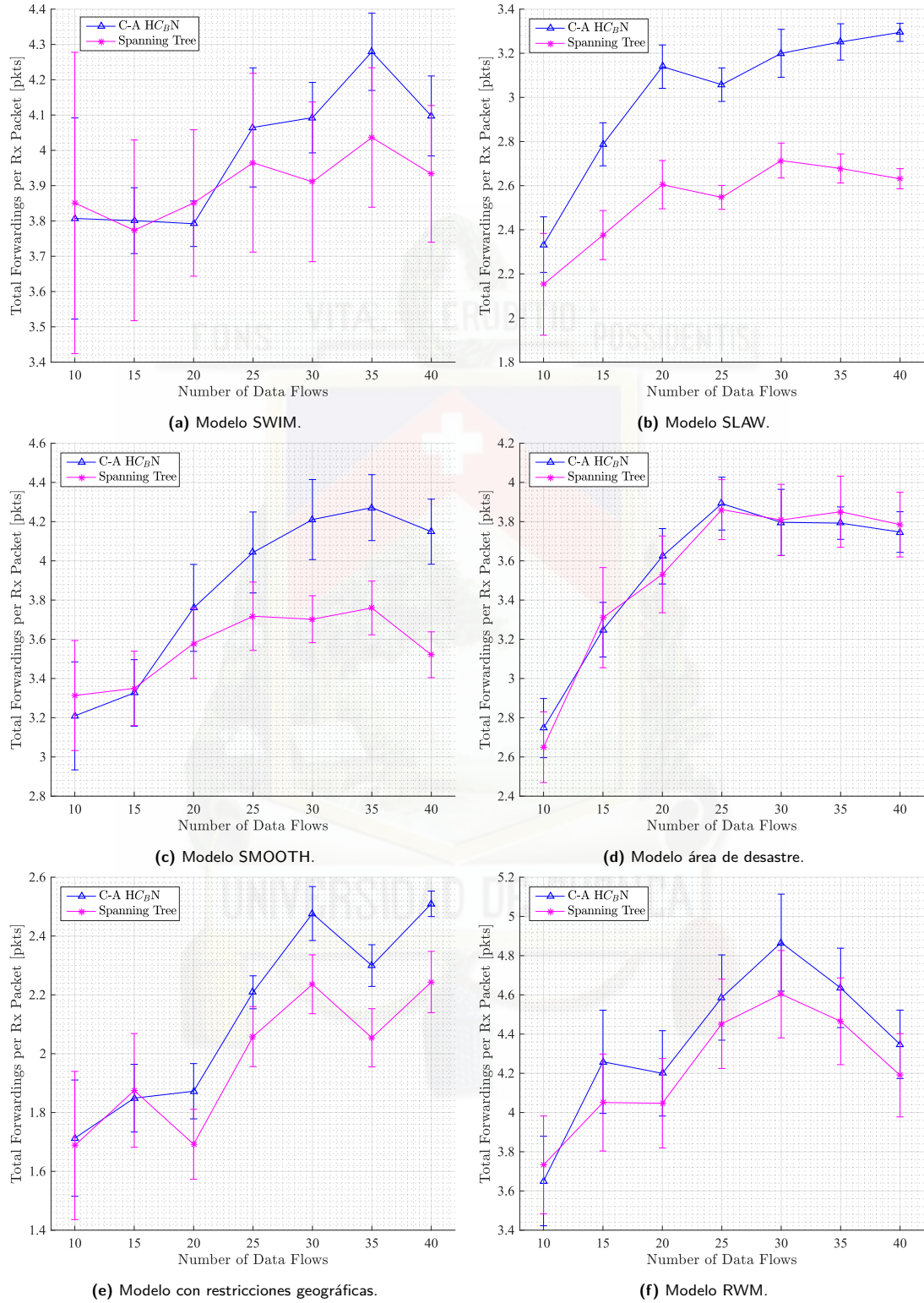


Figura 7.7: Resultados del total de datos reenviados por paquete recibido con éxito en función del numero de flujo para cada modelo de movilidad.

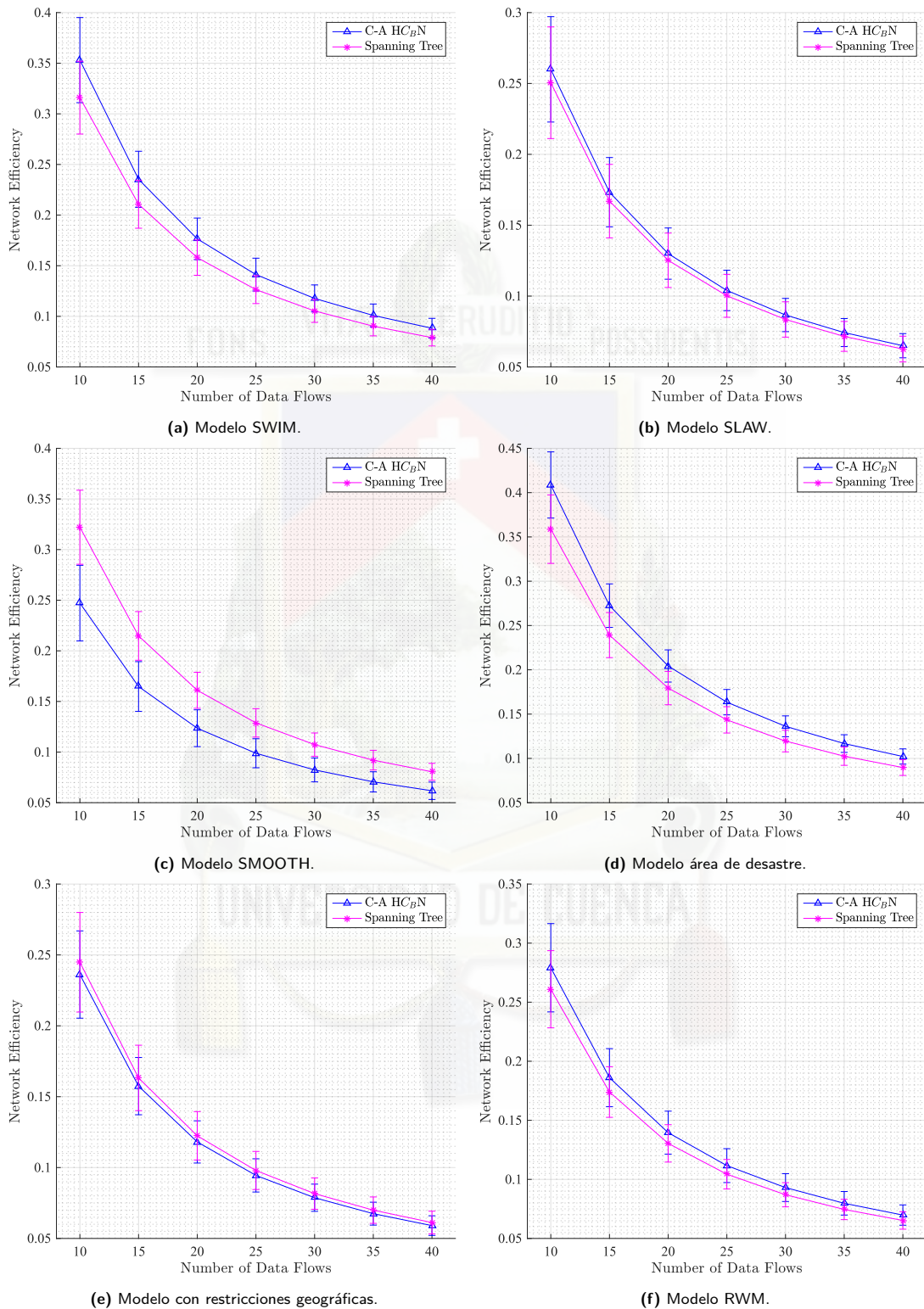


Figura 7.8: Resultados de la eficiencia de la red en términos PDR en función del numero de flujo para cada modelo de movilidad.

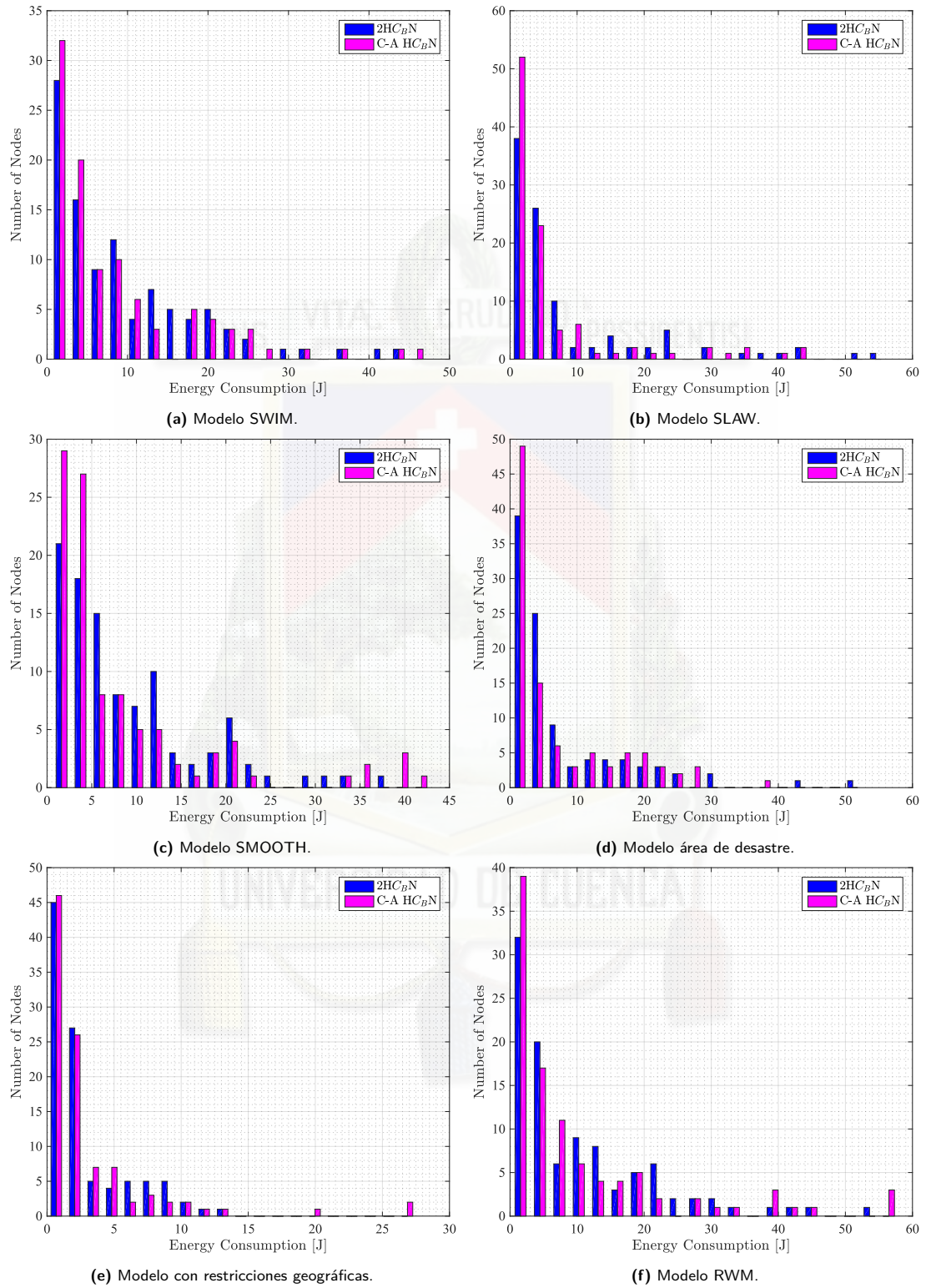


Figura 7.9: Histograma de consumo de energía por nodo para el caso de 20 flujos, resultados para cada modelo de movilidad.

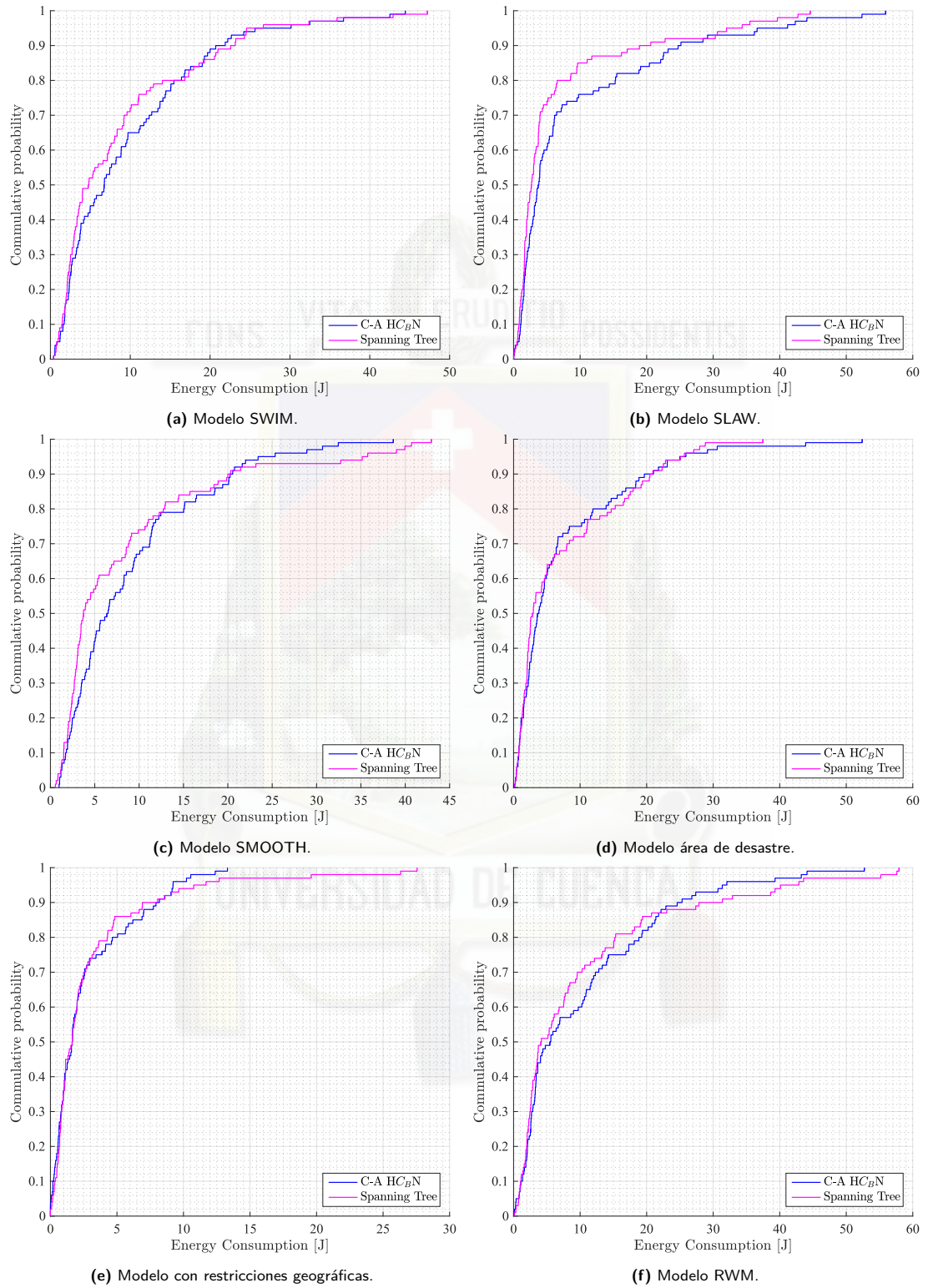


Figura 7.10: Función de distribución acumulada del consumo de energía por nodo para el caso de 20 flujos, resultados para cada modelo de movilidad.

Parámetro	Modelo	ST	(ST)– (C-A HC _B N)
Rate of routing managemnet messages	SWIM	91.02 %	3.39 %
	SLAW	93.16 %	6.61 %
	SMOOTH	86.85 %	4.16 %
	Disaster area	87.73 %	1.30 %
	Geographic Rest.	87.51 %	-0.59 %
	RWM	83.03 %	2.39 %
Total data forwardings	SWIM	47.95 %	6.42 %
	SLAW	33.97 %	6.86 %
	SMOOTH	38.14 %	13.92 %
	Disaster area	25.38 %	1.99 %
	Geographic Rest.	38.59 %	9.00 %
	RWM	27.93 %	-1.03 %
Successfully Rx packets	SWIM	3.30 %	-5.99 %
	SLAW	8.34 %	-4.74 %
	SMOOTH	8.76 %	4.33 %
	Disaster area	6.58 %	-0.53 %
	Geographic Rest.	9.16 %	-1.20 %
	RWM	3.68 %	-8.35 %
Total forwardings per Rx packet	SWIM	46.25 %	1.13 %
	SLAW	42.08 %	10.87 %
	SMOOTH	34.44 %	4.89 %
	Disaster area	29.55 %	0.26 %
	Geographic Rest.	37.21 %	4.20 %
	RWM	30.12 %	2.21 %
Network Efficiency	SWIM	3.13 %	-10.18 %
	SLAW	18.99 %	-2.94 %
	SMOOTH	20.29 %	8.54 %
	Disaster area	2.43 %	-8.52 %
	Geographic Rest.	16.35 %	3.07 %
	RWM	5.20 %	-7.59 %
Energy consumption	SWIM	47.86 %	1.68 %
	SLAW	45.27 %	20.55 %
	SMOOTH	30.96 %	4.53 %
	Disaster area	32.39 %	1.69 %
	Geographic Rest.	46.98 %	-5.52 %
	RWM	23.25 %	-8.11 %

Tabla 7.3: Resumen de porcentajes promedios de mejora del método ST con respecto a todos enrutadores y la diferencia con el método propuesto C-A HC_BN

7.5. Conclusiones

El mecanismo de control de topología basado en árboles de expansión consiste en la selección de nodos enrutadores en base a la reducción de la topología de red mediante algoritmos de árbol de expansión mínimo. Este método logra obtener mejores desempeños de las topologías resultantes que el método $C-A HC_BN$, consiguiendo conectividad completa de la red con menor cantidad de enrutadores y un menor número de cambios de estado.

Si bien la cantidad de nodos seleccionados en el caso de ST no es excesivamente bajo comparado a $C-A HC_BN$, la cantidad de cambios de estado decrece de manera muy significativa en el método de ST . Concretamente, disminuye aproximadamente a la mitad en los casos de $SWIM$ y Área de desastre, mientras que para los demás modelos disminuye a la tercera parte aproximadamente. Esta disminución de cambios de estado significa que las topologías obtenidas son más estables, lo cual es bueno desde el punto de vista de frecuencia de actualización de topología. Sin embargo, esto puede implicar un desbalance de carga y energía entre los nodos de la red. Existirán nodos que pasen la mayor parte de su tiempo como enrutadores por lo que consumirán mucha más energía que aquellos nodos que permanezcan como clientes.

Las simulaciones con tráfico en la red reflejan que existe una mejora tanto en la cantidad de paquetes de enrutamiento, reenvío de paquetes y por tanto consumo energético. Sin embargo la eficiencia de la red disminuye entre un 2.94 % y 10.18 % para la mayor parte de los modelos, con respecto al método $C-A HC_BN$.

En resumen, el método de selección de enrutadores ST consigue topologías más reducidas que $C-A HC_BN$. Sin embargo, debido al reducido número de MR , existe una mayor probabilidad que éstos lleguen a congestionarse, provocando una pérdida de paquetes que afectaría la eficiencia de la red. Adicionalmente, al tener un número reducido de nodos MR , la falla de uno de ellos tendría un impacto mucho más significativo y crítico en el desempeño de la red.



Capítulo 8

Conclusiones

En este capítulo se presentan las conclusiones generales del presente trabajo. Adicionalmente, se ilustran algunas posibles mejoras que se pueden implementar y los trabajos futuros.

8.1. Conclusiones

Las redes inalámbricas de malla están emergiendo cada vez más debido a las ventajas que traen consigo las redes descentralizadas, como la auto-organización y el bajo costo. En las redes de tipo malla, todos los dispositivos cumplen el rol de enrutadores de tráfico, formando una malla en la red. Pero el número excesivo de dispositivos móviles que pueden estar presentes en las redes espontáneas presentan problemas como la colisión de paquetes, interferencia y la sobrecarga general de la red debido a una señalización excesiva. Por tanto, se hace necesaria la implementación de mecanismos de control de topología en este tipo de redes.

Las métricas de centralidad ayudan a identificar los nodos más importantes de una red. De manera específica, la centralidad *Betweenness* representa la medida en la que un nodo tiene el control sobre la información que fluye en la red, mientras que la centralidad de puenteado (*Bridging Centrality*) identifica aquellos nodos que se encuentran entre dos regiones densamente conectadas. Por otra parte, las arquitecturas de redes de malla cliente se conforman principalmente por dispositivos transportados por personas, y consecuentemente adquieren el atributo de movilidad humana. Debido a las relaciones interpersonales y la manera en la que la sociedad interactúa en dichas redes, se forman comunidades en las que se evidencia que los nodos tienen una mejor interacción temporal y espacial entre los miembros de una misma comunidad.

Este trabajo propone un mecanismo de control de topología para redes de tipo malla que se basa en métricas de centralidad y en las estructuras comunitarias de las mismas. La idea fundamental del mecanismo propuesto es el de seleccionar aquellos nodos más relevantes en cada comunidad (identificados mediante la centralidad *Betweenness*) y aquellos elementos que comunican de mejor manera a éstas comunidades (identificados mediante la centralidad de puenteado). Estos elementos seleccionados formarán parte del núcleo de la red. El mecanismo fue nombrado **C-A HC_BN** (siglas correspondientes a: *Community-Aware Highest Betweenness Centrality Neighbor*) y se implementó y evaluó en 6 distintos escenarios cada uno con un modelo de movilidad distinto. Las topologías reducidas logran disminuir el número total de enlaces activos en la red en un 30.52 %-41.47 %.

Al aplicar tráfico en la red con el control de topología **C-A HC_BN**, se logra efectivamente un mejor desempeño de red. La cantidad de paquetes de enrutamiento es disminuido debido a la menor cantidad de enrutadores. La cantidad de paquetes reenviados también se reduce. Por tanto, los paquetes llegan a su destino en tiempos cortos y a través de rutas menos congestionadas. La cantidad de paquetes recibidos con éxito, aumenta debido a que existe menor cantidad de colisiones de paquetes e interferencia. Finalmente a nivel global la red experimenta un ahorro energético significativo.

Se compara el método propuesto de control de topología con el método **2HC_BN** de un trabajo

previo que no toma en cuenta la estructura comunitaria de la red. Aunque el índice de mensajes recibidos con éxito es similar en ambos casos, la cantidad de mensajes de enrutamiento y retransmisiones, disminuye con **C-A HC_BN**, provocando una mayor eficiencia de la red y un mayor ahorro energético. Esto concluye que efectivamente que la presente propuesta **C-A HC_BN** mejora el rendimiento de redes **WMNs** en escenarios con modelos de movilidad realistas.

Finalmente se compara el método de control de topología propuesto con un método basado en árboles de expansión (**ST**) **ST** logra obtener topologías más reducidas que el método **C-A HC_BN** consiguiendo conectividad completa de la red con menor cantidad de enrutadores y menor número de cambios de estado de los nodos. Sin embargo, debido al reducido número de **MR**, éstos se podrían llegar a congestionarse, provocando una pérdida de paquetes y mayor número de reenvío de paquetes lo que a la final se traduce como una menor eficiencia de la red. Por otra parte, el disminuido número de cambios de estado logrado con **ST** puede implicar el desgaste energético de nodos específicos ya que existirán algunos nodos que permanezcan demasiado tiempo como enrutadores. Por último, el método **ST** es más vulnerable ante fallos debido al reducido número de elementos que conforman el núcleo de la red.

8.2. Limitaciones, Mejoras y Trabajos Futuros

El método **C-A HC_BN** tiene un enfoque centralizado en este trabajo ya que las comunidades se detectan con el método de maximización de la modularidad y los cálculos de las centralidades han considerado a toda la red. Sin embargo, existen actualmente métodos de detección de comunidades [79] y cálculo de centralidad [103] que se ejecutan de manera distribuida y que podrían ser implementados y evaluados en esta propuesta.

Por otra parte, para el cálculo de la centralidad no se han considerado pesos en los nodos. Para la detección de comunidades se tomó en cuenta la métrica del protocolo de enrutamiento (AirTime Link metric) para el peso en las aristas, por lo que una mejora puede ser la introducción de nuevas métricas como: la energía restante del nodo, el tipo de dispositivo, la ubicación geográfica, etc.

El método **ST** logra obtener topologías más reducidas que los otros métodos analizados. Se podría aprovechar esta característica y complementar de alguna manera este esquema de selección de **MR** con algún otro esquema, con el fin de mejorar la eficiencia de la red con carga de tráfico. Adicionalmente, como trabajo futuro, se pretende evaluar este mecanismo para redes más grandes (mayor cantidad de nodos).





Apéndice A

Detección de comunidades



A.1. Number of communities detected

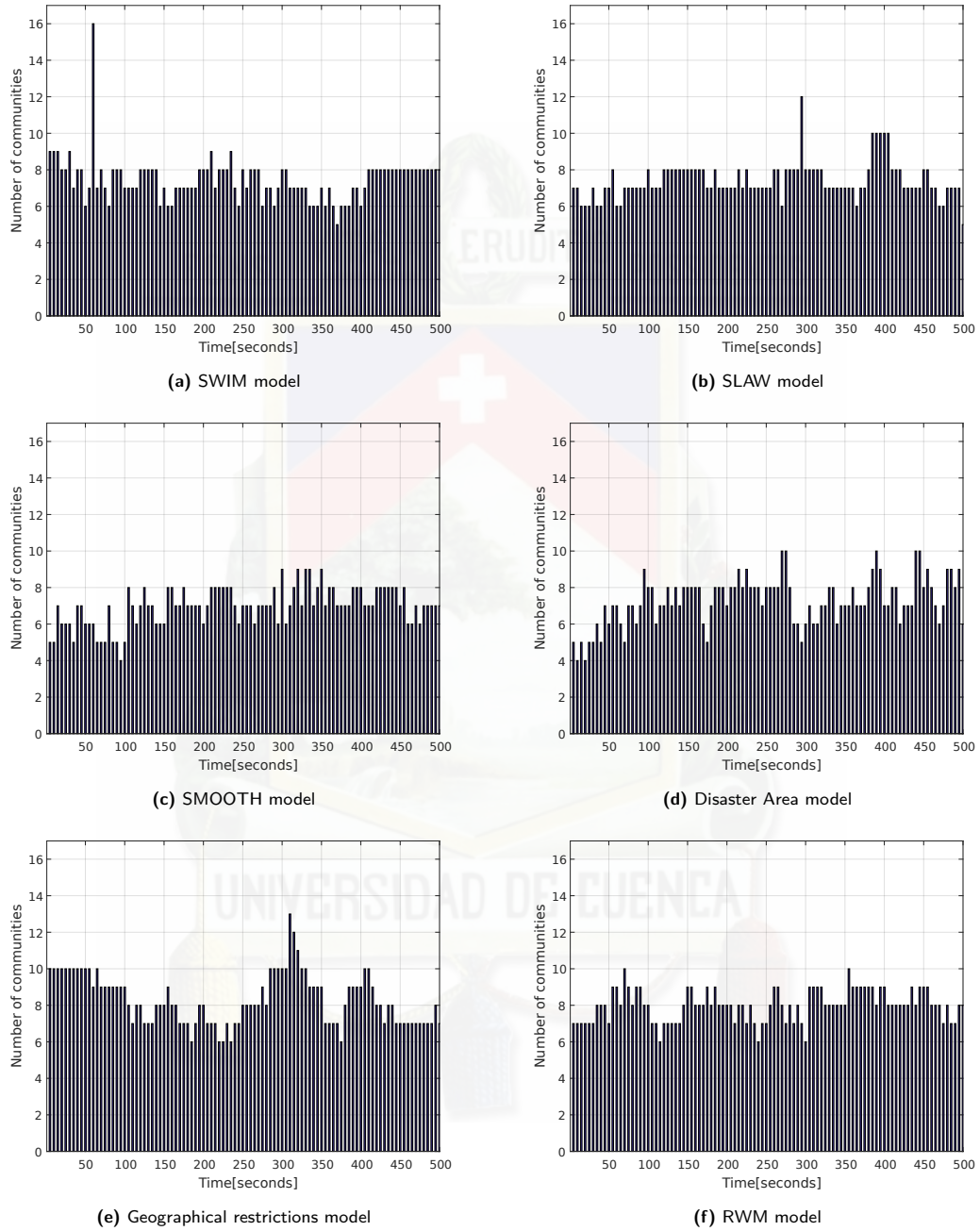


Figura A.1: Number of communities detected

A.2. Community Detection modularity obtained

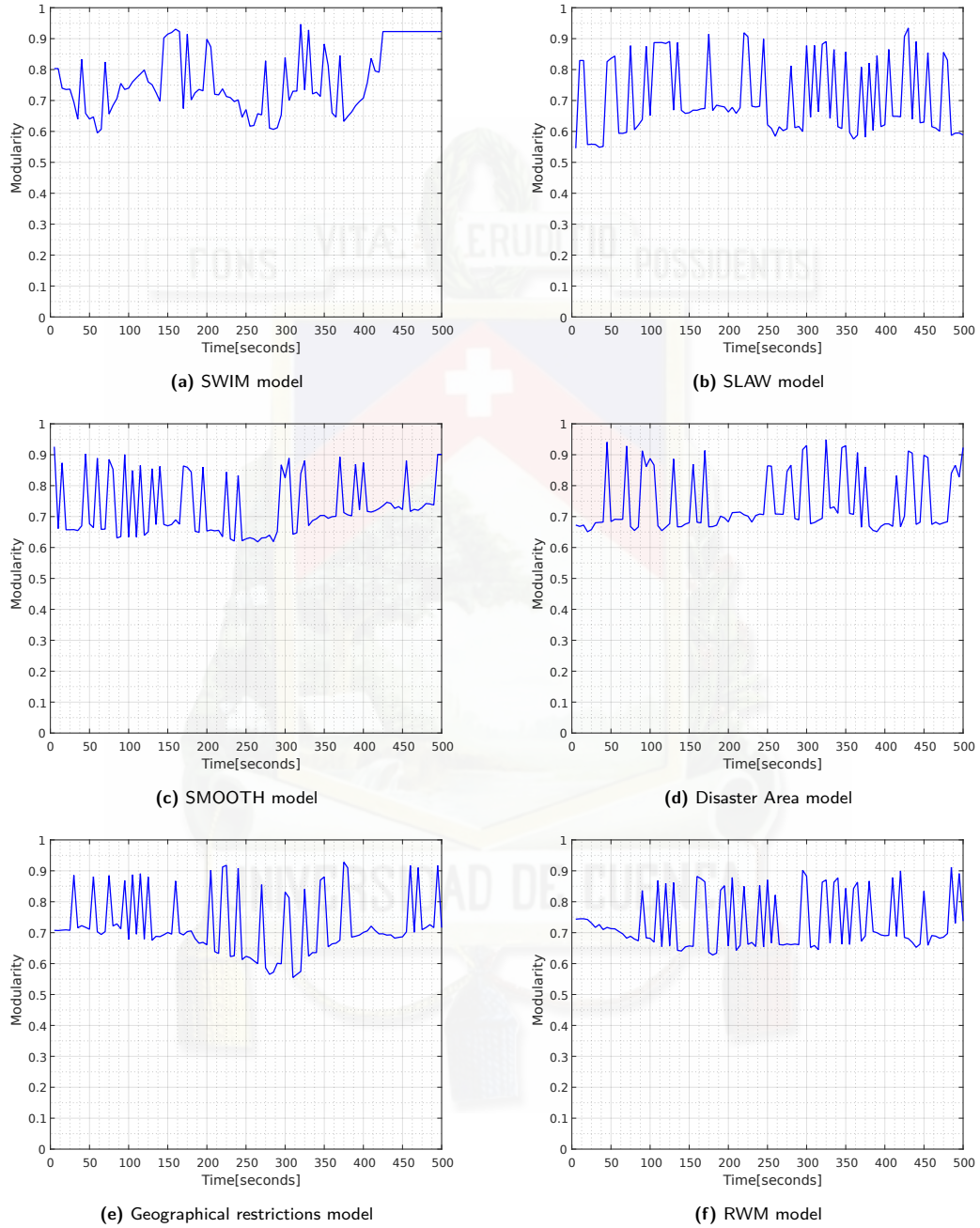


Figura A.2: Community Detection modularity obtained





Apéndice B

Scripts





B.1. Script principal de Octave (MAIN.m)

```
clc
clear all
close all
more off

%*****
%                               INPUT PARAMETERS
%*****

Sim='Sim_05';           %Simulation Folder
numNodes = 100;         %Number of nodes in simulation
Seconds=[5:5:500];      %Snapshots Interval

%*****
%                               READ MESH REPORTS AND GENERATE METIS
%*****

%OBTAIN ADJACENT AND METRIC MATRICES FROM MESH REPORTS
Adjacent_Matrix (Sim, numNodes, Seconds)

%GENERATE METIS GRAPH FILE FOR COMUNITY DETECTION INPUT
write_matrix_to_metis(Sim, numNodes, Seconds)

%EXECUTE COMMUNITY DETECTION
DetectCom (Sim, Seconds)

%*****
%                               INITIALIZATION
%*****

cont=1;
modularity=zeros(numel(Seconds),1); %Modularity
NumCom=zeros(numel(Seconds),1);     %Number of communities detected

NumLinks_Orig=zeros(numel(Seconds),1); %Number of edges (All-Routers)

NumRouters_Brid=zeros(numel(Seconds),1); %Number of selected routers (BRIDGING)
R_brid=zeros(numNodes,numel(Seconds)); %Routers vector (1 router, 0 no router, ...
-1 isolated) (BRIDGING)
NumLinks_Brid=zeros(numel(Seconds),1); %Number of edges (BRIDGING)

NumRouters_Back=zeros(numel(Seconds),1); %Number of selected routers (BACKBONE_WU)
R_back=zeros(numNodes,numel(Seconds)); %Routers vector (1 router, 0 no router, ...
-1 aislado) (BACKBONE_WU)
NumLinks_Back=zeros(numel(Seconds),1); %Number of edges (BACKBONE_WU)
```



```
for i=Seconds;

%*****
%
% IMPORTATIONS
%*****

%Import the coordinates from ns3 file generated in ns3
filename = ...
    strcat('/home/cristihan/ns-allinone-3.27/',Sim,'/MeshPos/Sec',num2str(i),'.txt');
delimiterIn = ' ';
headerlinesIn = 1;
aux = importdata(filename,delimiterIn,headerlinesIn);
coord=aux.data;
coord=[coord zeros(numNodes,1)]; %Add z component

%Import the Adjancy Matrix
filename = ...
    strcat('../Matrices/Simulaciones/',Sim,'/A_Matrix_Sec_',num2str(i),'.txt');
delimiterIn = ' ';
headerlinesIn = 0;
aux = importdata(filename,delimiterIn,headerlinesIn);
A=aux;
NumLinks_Orig(cont)=sum(sum(A));

%Import the modularity from community detection
filename = ...
    strcat('../Communities/',Sim,'/Complete_Weight_Graph_Sec_',num2str(i),'.graph_modularity');
delimiterIn = ' ';
headerlinesIn = 0;
aux = importdata(filename,delimiterIn,headerlinesIn);
modularity(cont)=aux;

%Import Communities from community detection
filename= ...
    strcat('../Communities/',Sim,'/Complete_Weight_Graph_Sec_',num2str(i),'.graph_clustInfo');
delimiterIn = ' ';
communities = importdata(filename,delimiterIn); %values between -1 and NumCom-1
communities = communities + 1; %values between 0 and NUmCom ...
    (0 means no community)
NumCom(cont)=max(communities);

%*****
%
% ROUTERS SELECTING
%*****

%Select two most central nodes in each community, R1 contains all the routers ...
    selected
centrality=betweenness_bin(A);
R1=zeros(numNodes,1);
for community=[1:1:NumCom(cont)]
    Aprueba=zeros(numNodes,numNodes);
```




```
I=find(communities==community); %Indices of i community nodes
Aprueba(I,:)=A(I,:);
Aprueba(:,I)=A(:,I);
[Rprueba] = MostCentralCom (Aprueba,2,centrality,I);
R1(Rprueba)=1;
end
clear I Rprueba

%Obtain reduced Graph obtained by bridging method (bridging centrality)
[R_brid(:,cont),A_reduced_brid,backbone] = ReducedTopologyBridging (A,0.37,R1);
NumLinks_Brid(cont)=sum(sum(A_reduced_brid));
topRouters=find(R_brid(:,cont)==1);
NumRouters_Brid(cont)=numel(topRouters);
disp(strcat('Number of Routers Selected with bridging for ...
second_',num2str(i),' : ',num2str(NumRouters_Brid(cont))))
clear R1

%Obtain reduced Graph obtained by backbone method (backbone_wu)
[R_back(:,cont),A_reduced_back,backbone_back] = ReducedTopologyBackbone (A);
NumLinks_Back(cont)=sum(sum(A_reduced_back));
topRouters=find(R_back(:,cont)==1);
NumRouters_Back(cont)=numel(topRouters);
disp(strcat('Number of Routers Selected with backbone for ...
second_',num2str(i),' : ',num2str(NumRouters_Back(cont))))

%*****
%                               COUNTERS AND CLEAR VARIABLES
%*****

cont=cont+1;
backbone=[];
coord=[];
end

%*****
%                               SHOW OUTPUT DATA
%*****

NumCom
modularity
NumRouters_Brid
R_brid
NumRouters_Back
R_back
NumLinks_Orig
NumLinks_Brid
NumLinks_Back
```



B.2. Script principal de ns-3 (MAIN.cc)

[illegible]



```
std::string m_stack;
std::string m_root;
double PathToRootTimeout;
double RootInterval;
int Nroots;
uint32_t m_maxPacketCount;
int nodeNum;
std::string traceFile;
std::string outFolder;
int m_runN;
int Flows;
int num_sim;
int Snapshot;

NodeContainer nodes; // List of network nodes
NetDeviceContainer meshDevices; // List of all mesh point devices
Ipv4InterfaceContainer interfaces; //Addresses of interfaces:
MeshHelper mesh; // MeshHelper. Report is not static methods

private:
    // Create nodes and setup their mobility
    void CreateNodes ();
    // Install internet m_stack on nodes
    void InstallInternetStack ();
    // Install applications
    void InstallApplication ();
    // Print mesh devices diagnostics
    void Report ();
    void ReportTime ();
    void ReportNode ();
};

MeshTest::MeshTest () :
    m_randomStart (0.5),
    m_totalTime (385),
    m_packetInterval (0.02),
    m_packetSize (512),
    m_nIfaces (1),
    m_chan (true),
    m_pcap (false),
    m_stack ("ns3::Dot11sStack"),
    m_root ("ff:ff:ff:ff:ff:ff"),
    PathToRootTimeout (5120000), //DEFAULT VALUE
    RootInterval (2048000), //DEFAULT VALUE
    m_maxPacketCount (10000),
    nodeNum (100),
    traceFile ("TraceFile.txt"),
    outFolder ("Tesis/Out_Proac_AllR") //<=====CARPETA DE SALIDA
{
```



```
}

// *****
//                                     MeshTest::Configure
// *****
void
MeshTest::Configure (int argc, char *argv[])
{
    CommandLine cmd;

    cmd.AddValue ("start", "Maximum random start delay, seconds. [0.1 s]", ...
        m_randomStart);
    cmd.AddValue ("time", "Simulation time, seconds [500 s]", m_totalTime);
    cmd.AddValue ("packet-interval", "Interval between packets in UDP ping, ...
        seconds [0.001 s]", m_packetInterval);
    cmd.AddValue ("packet-size", "Size of packets in UDP ping", m_packetSize);
    cmd.AddValue ("interfaces", "Number of radio interfaces used by each mesh ...
        point. [1]", m_nIfaces);
    cmd.AddValue ("channels", "Use different frequency channels for different ...
        interfaces. [0]", m_chan);
    cmd.AddValue ("pcap", "Enable PCAP traces on interfaces. [0]", m_pcap);
    cmd.AddValue ("stack", "Type of protocol stack. ns3::Dot11sStack by default", ...
        m_stack);

    cmd.AddValue ("root1", "Mac address of root1 mesh point in HWMP", m_root1);
    cmd.AddValue ("root2", "Mac address of root2 mesh point in HWMP", m_root2);
    cmd.AddValue ("root3", "Mac address of root3 mesh point in HWMP", m_root3);
    cmd.AddValue ("PTRtimeout", "Lifetime of proactive routing information", ...
        PathToRootTimeout);
    cmd.AddValue ("Rinterval", "Interval between two successive proactive PREQs", ...
        RootInterval);
    cmd.AddValue ("Nroots", "Number of roots in network", Nroots);
    cmd.AddValue ("maxPacketCount", "Maximum TX packets", m_maxPacketCount);
    cmd.AddValue ("nodeNum", "Number of nodes", nodeNum);
    cmd.AddValue ("traceFile", "Ns2 movement trace file", traceFile);
    cmd.AddValue ("outFolder", "Output folder for simulation results", outFolder);

    cmd.AddValue ("m_runN", "Number of run execution ", m_runN);
    cmd.AddValue ("Flows", "Number of flows", Flows);
    cmd.AddValue ("num_sim", "number of actual Snapshot", num_sim );
    cmd.AddValue ("Snapshot", "number of actual Snapshot", Snapshot );

    cmd.Parse (argc, argv);
    NS_LOG_DEBUG ("Simulation time: " << m_totalTime << " s");
}

// *****
//                                     MeshTest::CreateNodes
// *****
```



```
void
MeshTest::CreateNodes ()
{

    // Create Ns2MobilityHelper with the specified trace log file as parameter
    Ns2MobilityHelper ns2 = Ns2MobilityHelper (traceFile);
    nodes.Create (nodeNum);
    ns2.Install (); // configure movements for each node, while reading trace file

    // Configure YansWifiChannel
    YansWifiPhyHelper wifiPhy = YansWifiPhyHelper::Default ();
    YansWifiChannelHelper wifiChannel = YansWifiChannelHelper::Default ();
    wifiPhy.SetChannel (wifiChannel.Create ());

    mesh = MeshHelper::Default ();
    if (!Mac48Address (m_root.c_str()).IsBroadcast ())
    {
        mesh.SetStackInstaller (m_stack, "Root", Mac48AddressValue (Mac48Address ...
            (m_root.c_str ()))));
    }
    else
    {
        //If root is not set, we do not use "Root" attribute, because it
        //is specified only for 11s
        mesh.SetStackInstaller (m_stack);
    }
    if (m_chan)
    {
        mesh.SetSpreadInterfaceChannels (MeshHelper::SPREAD_CHANNELS);
    }
    else
    {
        mesh.SetSpreadInterfaceChannels (MeshHelper::ZERO_CHANNEL);
    }
    mesh.SetMacType ("RandomStart", TimeValue (Seconds (m_randomStart)));
    mesh.SetNumberOfInterfaces (m_nIfaces); // Set number of interfaces - default is ...
        single-interface mesh point
    meshDevices = mesh.Install (wifiPhy, nodes); // Install protocols and return ...
        container if MeshPointDevices

    std::ostringstream ossss;
    for (int i=0; i<100; i++)
    {
        if (Routers [i] == 0) {
            ossss.str("");
            ossss << "/NodeList/" << i << ...
                "/DeviceList/0/$ns3::MeshPointDevice/RoutingProtocol/$ns3::dot11s::HwmpProtocol/IsRouter"
            Config::Set (ossss.str(), BooleanValue (false));
        }
    }
}
```



```
if (m_pcap)
    wifiPhy.EnablePcapAll (std::string ("mp-"));
}

//*****
//                               MeshTest::InstallInternetStack
//*****
void
MeshTest::InstallInternetStack ()
{
    InternetStackHelper internetStack;
    internetStack.Install (nodes);
    Ipv4AddressHelper address;
    address.SetBase ("10.1.1.0", "255.255.255.0");
    interfaces = address.Assign (meshDevices);
}

//*****
//                               MeshTest::InstallApplication
//*****
void
MeshTest::InstallApplication ()
{
    std::cout << "*****Flow: "<< Flows << " --- Run: " << ...
        m_runN << "*****\n";
    include "TrafficFlowDirectory"
    std::cout << "\n";
}

//*****
//                               MeshTest::Run()
//*****
int
MeshTest::Run ()
{
    CreateNodes ();
    InstallInternetStack ();
    InstallApplication ();

    //schedulers
    for (int i=1; i<=m_totalTime; i=i+1) {
        Simulator::Schedule (Seconds (i), &MeshTest::ReportTime, this); // Solo para ...
            mostrar el tiempo virtual de simulacion
    }
    Simulator::Schedule (Seconds (m_totalTime), &MeshTest::ReportNode, this); // ...
        Reporte de c/nodo al final de la simulacion
    Simulator::Stop (Seconds (m_totalTime)); //fijo el tiempo en el que debe detenerse

    //FlowMonitor
    Ptr<FlowMonitor> flowMon;
```



```
FlowMonitorHelper flowMonHelper;
flowMon = flowMonHelper.InstallAll();

//RUN
Simulator::Run ();
std::ostringstream oss;
oss << outFolder << "/Sim0" << num_sim << "/Snap_" << Snapshot << "/Flow_Mon/R" ...
    << Nroots << "F" << Flows << "T" << PathToRootTimeout/10000 << "0000_Run" ...
    << m_runN << ".xml";
flowMon->SerializeToXmlFile(oss.str(), true, true);
Simulator::Destroy ();

return 0;
}

//*****
//                               MeshTest::ReportTime
//*****
void
MeshTest::ReportTime ()
{
    std::cerr << "Virtual time simulation: " << Simulator::Now().GetSeconds() << ...
        "\n";
}

//*****
//                               MeshTest::ReportNode
//*****
void
MeshTest::ReportNode ()
{
    unsigned n (0);
    for (NetDeviceContainer::Iterator i = meshDevices.Begin (); i != ...
        meshDevices.End (); ++i, ++n)
    {
        std::ostringstream os;
        os << outFolder << "/Sim0" << num_sim << "/Snap_" << Snapshot << ...
            "/Mesh_Reports/R" << Nroots << "F" << Flows << "T" << ...
            PathToRootTimeout/10000 << "0000/Run" << m_runN << "/Node" << n << ...
            ".xml";
        std::ofstream of;
        of.open (os.str ().c_str ());
        if (!of.is_open ())
        {
            std::cerr << "Error: Can't open file " << os.str () << "\n";
            return;
        }
        mesh.Report (*i, of);
        of.close ();
    }
}
```




```
std::cerr << "Printing mesh point devices at time #" << ...
    Simulator::Now().GetSeconds() << "\n";
std::cerr << "Node Reporte Successful\n";
}

//*****
//                               main
//*****
int
main (int argc, char *argv[])
{
    MeshTest t;
    t.Configure (argc, argv);
    return t.Run ();
}
```



Bibliografía

- [1] A. M. Vázquez Rodas, “Contribution to the improvement of the performance of wireless mesh networks providing real time services,” Ph.D. dissertation, Universitat Politècnica de Catalunya, 2015.
- [2] I. F. Akyildiz, X. Wang, y W. Wang, “Wireless mesh networks: a survey,” *Computer networks*, vol. 47, num. 4, pp. 445–487, 2005.
- [3] G. Aloï, M. Di Felice, V. Loscrì, P. Pace, y G. Ruggeri, “Spontaneous smartphone networks as a user-centric solution for the future internet,” *IEEE Communications Magazine*, vol. 52, num. 12, pp. 26–33, 2014.
- [4] W. Z. Khan, Y. Xiang, M. Y. Aalsalem, y Q. Arshad, “Mobile phone sensing systems: A survey,” *IEEE Communications Surveys & Tutorials*, vol. 15, num. 1, pp. 402–427, 2013.
- [5] A. A. Aziz, Y. A. Sekercioglu, P. Fitzpatrick, y M. Ivanovich, “A survey on distributed topology control techniques for extending the lifetime of battery powered wireless sensor networks,” *IEEE communications surveys & tutorials*, vol. 15, num. 1, pp. 121–144, 2013.
- [6] D. B. Johnson y D. A. Maltz, “Dynamic source routing in ad hoc wireless networks,” *Mobile computing*, pp. 153–181, 1996.
- [7] D. Karamshuk, C. Boldrini, M. Conti, y A. Passarella, “Human mobility models for opportunistic networks,” *IEEE Communications Magazine*, vol. 49, num. 12, pp. 157–165, 2011.
- [8] A. Mei y J. Stefa, “Swim: A simple model to generate small mobile worlds,” in *INFOCOM 2009, IEEE*. IEEE, 2009, pp. 2106–2113.
- [9] K. Lee, S. Hong, S. J. Kim, I. Rhee, y S. Chong, “Slaw: A new mobility model for human walks,” in *INFOCOM 2009, IEEE*. IEEE, 2009, pp. 855–863.
- [10] A. Munjal, T. Camp, y W. C. Navidi, “Smooth: a simple way to model human walks,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 14, num. 4, pp. 34–36, 2011.



- [11] N. Aschenbruck, E. Gerhards-Padilla, M. Gerharz, M. Frank, y P. Martini, “Modelling mobility in disaster area scenarios,” in *Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems*. ACM, 2007, pp. 4–12.
- [12] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, y D. Parisi, “Defining and identifying communities in networks,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, num. 9, pp. 2658–2663, 2004.
- [13] M. Girvan y M. E. Newman, “Community structure in social and biological networks,” *Proceedings of the national academy of sciences*, vol. 99, num. 12, pp. 7821–7826, 2002.
- [14] T. Leighton y S. Rao, “Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms,” *Journal of the ACM (JACM)*, vol. 46, num. 6, pp. 787–832, 1999.
- [15] D. A. Spielman y S.-H. Teng, “Spectral partitioning works: Planar graphs and finite element meshes,” *Linear Algebra and its Applications*, vol. 421, num. 2-3, pp. 284–305, 2007.
- [16] G. Karypis y V. Kumar, “A fast and high quality multilevel scheme for partitioning irregular graphs,” *SIAM Journal on scientific Computing*, vol. 20, num. 1, pp. 359–392, 1998.
- [17] I. S. Dhillon, Y. Guan, y B. Kulis, “Weighted graph cuts without eigenvectors a multilevel approach,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, num. 11, 2007.
- [18] H. Lu, M. Halappanavar, D. Chavarria-Miranda, A. H. Gebremedhin, A. Panyala, y A. Kalyanaraman, “Algorithms for balanced graph colorings with applications in parallel computing,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, num. 5, pp. 1240–1256, 2017.
- [19] M. E. Newman y M. Girvan, “Finding and evaluating community structure in networks,” *Physical review E*, vol. 69, num. 2, p. 026113, 2004.
- [20] M. E. Newman, “Fast algorithm for detecting community structure in networks,” *Physical review E*, vol. 69, num. 6, p. 066133, 2004.
- [21] J. Leskovec, K. J. Lang, y M. Mahoney, “Empirical comparison of algorithms for network community detection,” in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 631–640.
- [22] S. Fortunato y D. Hric, “Community detection in networks: A user guide,” *Physics Reports*, vol. 659, pp. 1–44, 2016.

- [23] T. Qiu, B. Chen, A. K. Sangaiah, J. Ma, y R. Huang, "A survey of mobile social networks: Applications, social characteristics, and challenges," *IEEE Systems Journal*, 2017.
- [24] ns-3. [En línea]. Disponible: <https://www.nsnam.org/>
- [25] BonnMotion - a mobility scenario generation and analysis tool. [En línea]. Disponible: <http://sys.cs.uos.de/bonnmotion/>
- [26] D. Krajzewicz, J. Erdmann, M. Behrisch, y L. Bieker, "Recent development and applications of SUMO - Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, vol. 5, num. 3&4, pp. 128–138, December 2012.
- [27] Gephi - the open graph viz platform. [En línea]. Disponible: <https://gephi.org/>
- [28] GNU octave. [En línea]. Disponible: <https://www.gnu.org/software/octave/>
- [29] Welcome to python.org. [En línea]. Disponible: <https://www.python.org/>
- [30] I. . L. S. Committee y otros, "Ieee standard for information technology-telecommunication and information exchange between systems-local and metropolitan area networks-specific requirements part11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment1: Radio resource measurement of wireless lans," http://standards.ieee.org/getieee802/download/802.11_n-2009, 2009.
- [31] J. Wu y I. Stojmenovic, "Ad hoc networks," *Computer*, vol. 37, num. 2, pp. 29–31, 2004.
- [32] K. Andreev y P. Boyko, "Ieee 802.11 s mesh networking ns-3 model," in *Workshop on ns3*, vol. 43, 2010.
- [33] "IEEE 802.11-2016 - IEEE Standard for Information technology—Telecommunications and information exchange between systems Local and metropolitan area networks—Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications." [En línea]. Disponible: <https://standards.ieee.org/findstds/standard/802.11-2016.html>
- [34] X. Wang y A. O. Lim, "Ieee 802.11 s wireless mesh networks: Framework and challenges," *Ad Hoc Networks*, vol. 6, num. 6, pp. 970–984, 2008.
- [35] P. Santi, "Topology control in wireless ad hoc and sensor networks," *ACM computing surveys (CSUR)*, vol. 37, num. 2, pp. 164–194, 2005.
- [36] R. Rajaraman, "Topology control and routing in ad hoc networks: A survey," *ACM SIGACT News*, vol. 33, num. 2, pp. 60–73, 2002.
- [37] N. Li, J. C. Hou, y L. Sha, "Design and analysis of an mst-based topology control algorithm," *IEEE Transactions on Wireless Communications*, vol. 4, num. 3, pp. 1195–1206, 2005.



- [38] R. Wattenhofer, L. Li, P. Bahl, y Y.-M. Wang, "Distributed topology control for power efficient operation in multihop wireless ad hoc networks," in *INFOCOM 2001. Twentieth annual joint conference of the IEEE computer and communications societies. Proceedings. IEEE*, vol. 3. IEEE, 2001, pp. 1388–1397.
- [39] R. Ramanathan y R. Rosales-Hain, "Topology control of multihop wireless networks using transmit power adjustment," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2. IEEE, 2000, pp. 404–413.
- [40] L. Bao y J. J. Garcia-Luna-Aceves, "Topology management in ad hoc networks," in *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*. ACM, 2003, pp. 129–140.
- [41] L. Jia, R. Rajaraman, y T. Suel, "An efficient distributed algorithm for constructing small dominating sets," *Distributed Computing*, vol. 15, num. 4, pp. 193–205, 2002.
- [42] M. Shlesinger, B. West, y J. Klafter, "Lévy dynamics of enhanced diffusion: Application to turbulence," *Physical Review Letters*, vol. 58, num. 11, p. 1100, 1987.
- [43] S. Benhamou, "How many animals really do the levy walk?" *Ecology*, vol. 88, num. 8, pp. 1962–1969, 2007.
- [44] A. M. Edwards, R. A. Phillips, N. W. Watkins, M. P. Freeman, E. J. Murphy, V. Afanasyev, S. V. Buldyrev, M. G. da Luz, E. P. Raposo, H. E. Stanley y otros, "Revisiting lévy flight search patterns of wandering albatrosses, bumblebees and deer," *Nature*, vol. 449, num. 7165, pp. 1044–1048, 2007.
- [45] I. Rhee, M. Shin, S. Hong, K. Lee, S. J. Kim, y S. Chong, "On the levy-walk nature of human mobility," *IEEE/ACM transactions on networking (TON)*, vol. 19, num. 3, pp. 630–643, 2011.
- [46] T. Karagiannis, J.-Y. Le Boudec, y M. Vojnovic, "Power law and exponential decay of intercontact times between mobile devices," *IEEE Transactions on Mobile Computing*, vol. 9, num. 10, pp. 1377–1390, 2010.
- [47] H. Cai y D. Y. Eun, "Crossing Over the Bounded Domain: From Exponential to Power-Law Intermeeting Time in Mobile Ad Hoc Networks," *IEEE/ACM Transactions on Networking*, vol. 17, num. 5, pp. 1578–1591, Oct. 2009. [En línea]. Disponible: <http://ieeexplore.ieee.org/document/5175360/>
- [48] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, y C. Diot, "Pocket switched networks and human mobility in conference environments." ACM Press, 2005, pp. 244–251. [En línea]. Disponible: <http://portal.acm.org/citation.cfm?doid=1080139.1080142>



- [49] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, y J. Scott, "Impact of Human Mobility on Opportunistic Forwarding Algorithms," *IEEE Transactions on Mobile Computing*, vol. 6, num. 6, pp. 606–620, Jun. 2007. [En línea]. Disponible: <http://ieeexplore.ieee.org/document/4161914/>
- [50] J. Leguay, A. Lindgren, J. Scott, T. Friedman, y J. Crowcroft, "Opportunistic content distribution in an urban setting." ACM Press, 2006, pp. 205–212. [En línea]. Disponible: <http://portal.acm.org/citation.cfm?doid=1162654.1162657>
- [51] S. Kosta, A. Mei, y J. Stefa, "Large-scale synthetic social mobile networks with SWIM," *IEEE Transactions on Mobile Computing*, vol. 13, num. 1, pp. 116–129, 2014.
- [52] "SWIM: Small World In Motion - a mobility model for human movements."
- [53] D. Brockmann, L. Hufnagel, y T. Geisel, "The scaling laws of human travel," *Nature*, vol. 439, num. 7075, pp. 462–465, 2006.
- [54] M. C. Gonzalez, C. A. Hidalgo, y A.-L. Barabasi, "Understanding individual human mobility patterns," *Nature*, vol. 453, num. 7196, pp. 779–782, 2008.
- [55] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, y J. Scott, "Impact of human mobility on opportunistic forwarding algorithms," *IEEE Transactions on Mobile Computing*, vol. 6, num. 6, 2007.
- [56] I. Norros, P. Mannersalo, y J. L. Wang, "Simulation of fractional brownian motion with conditionalized random midpoint displacement," *Advances in Performance Analysis*, vol. 2, num. 1, pp. 77–101, 1999.
- [57] "Human Mobility Models Download (TLW, SLAW) | Networking Research Lab." [En línea]. Disponible: <https://research.csc.ncsu.edu/netsrv/?q=content/human-mobility-models-download-tlw-slaw>
- [58] A. Munjal, T. Camp, y W. C. Navidi, "Constructing rigorous manet simulation scenarios with realistic mobility," in *Wireless Conference (EW), 2010 European*. IEEE, 2010, pp. 817–824.
- [59] "Toilers- Colorado School of Mines- SMOOTH." [En línea]. Disponible: <http://toilers.mines.edu/Code/smooth.html>
- [60] N. Aschenbruck, E. Gerhards-Padilla, y P. Martini, "Modeling mobility in disaster area scenarios," *Performance Evaluation*, vol. 66, num. 12, pp. 773–790, 2009.
- [61] M. Schwamborn y N. Aschenbruck, "Introducing geographic restrictions to the slaw human mobility model," in *Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2013 IEEE 21st International Symposium on*. IEEE, 2013, pp. 264–272.

- [62] OpenStreetMap. [En línea]. Disponible: <https://www.openstreetmap.org/>
- [63] S. I. Souravlas y A. Sifaleras, "On the detection of overlapped network communities via weight redistributions," in *GeNeDis 2016*. Springer, 2017, pp. 205–214.
- [64] S. Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, num. 3-5, pp. 75–174, 2010.
- [65] M. A. Tabarzag y A. Hamzeh, "A heuristic local community detection method (hlcd)," *Applied Intelligence*, vol. 46, num. 1, pp. 62–78, 2017.
- [66] Z. Lu, X. Sun, Y. Wen, G. Cao, y T. La Porta, "Algorithms and applications for community detection in weighted networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, num. 11, pp. 2916–2926, 2015.
- [67] D. Katsaros, N. Dimokas, y L. Tassiulas, "Social network analysis concepts in the design of wireless ad hoc network protocols," *IEEE network*, vol. 24, num. 6, 2010.
- [68] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social networks*, vol. 1, num. 3, pp. 215–239, 1978.
- [69] L. Page, S. Brin, R. Motwani, y T. Winograd, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Tech. Rep., 1999.
- [70] M. Ramanathan, A. Zhang, Y.-r. Cho, y W. Hwang, "Bridging centrality: identifying bridging nodes in scale-free networks," 2006.
- [71] J. Huang, Q. Duan, C.-C. Xing, y H. Wang, "Topology control for building a large-scale and energy-efficient internet of things," *IEEE Wireless Communications*, vol. 24, num. 1, pp. 67–73, 2017.
- [72] G. Ma, Y. Yang, X. Qiu, Z. Gao, y H. Li, "Fault-tolerant topology control for heterogeneous wireless sensor networks using multi-routing tree," in *Integrated Network and Service Management (IM), 2017 IFIP/IEEE Symposium on*. IEEE, 2017, pp. 620–623.
- [73] M. M. Qirtas, Y. Faheem, y M. H. Rehmani, "Throwboxes in delay tolerant networks: A survey of placement strategies, buffering capacity, and mobility models," *Journal of Network and Computer Applications*, vol. 91, pp. 89–103, 2017.
- [74] C.-C. Lin, P.-T. Tseng, T.-Y. Wu, y D.-J. Deng, "Social-aware dynamic router node placement in wireless mesh networks," *Wireless Networks*, vol. 22, num. 4, pp. 1235–1250, 2016.
- [75] M. Kas, S. Appala, C. Wang, K. M. Carley, L. R. Carley, y O. K. Tonguz, "What if wireless routers were social? approaching wireless mesh networks from a social networks perspective," *IEEE Wireless Communications*, vol. 19, num. 6, 2012.



- [76] N. Meghanathan, "Use of centrality metrics to determine connected dominating sets for real-world network graphs," in *Information Technology-New Generations (ITNG), 2015 12th International Conference on*. IEEE, 2015, pp. 243–248.
- [77] A. Vázquez-Rodas y J. Luis, "A centrality-based topology control protocol for wireless mesh networks," *Ad Hoc Networks*, vol. 24, pp. 34–54, 2015.
- [78] P. Hui, J. Crowcroft, y E. Yoneki, "Bubble rap: Social-based forwarding in delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 10, num. 11, pp. 1576–1589, 2011.
- [79] P. Hui, E. Yoneki, S. Y. Chan, y J. Crowcroft, "Distributed community detection in delay tolerant networks," in *Proceedings of 2nd ACM/IEEE international workshop on Mobility in the evolving internet architecture*. ACM, 2007, p. 7.
- [80] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, y E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, vol. 2008, num. 10, p. P10008, 2008.
- [81] H. Lu, M. Halappanavar, y A. Kalyanaraman, "Parallel heuristics for scalable community detection," *Parallel Computing*, vol. 47, pp. 19–37, 2015.
- [82] The 3-clause BSD license | open source initiative. [En línea]. Disponible: <https://opensource.org/licenses/BSD-3-Clause>
- [83] S. P. Borgatti, "Identifying sets of key players in a social network," *Computational & Mathematical Organization Theory*, vol. 12, num. 1, pp. 21–34, 2006.
- [84] "BonnMotion - A mobility scenario generation and analysis tool." [En línea]. Disponible: <http://sys.cs.uos.de/bonnmotion/>
- [85] N. Aschenbruck, R. Ernst, E. Gerhards-Padilla, y M. Schwamborn, "BonnMotion: a mobility scenario generation and analysis tool." ICST, 2010. [En línea]. Disponible: <http://eudl.eu/doi/10.4108/ICST.SIMUTOOLS2010.8684>
- [86] "OpenStreetMap." [En línea]. Disponible: <https://www.openstreetmap.org/>
- [87] METIS GRAPH files. [En línea]. Disponible: http://people.sc.fsu.edu/~jburkardt/data/metis_graph/metis_graph.html
- [88] S. Kintali, "Betweenness centrality: Algorithms and lower bounds," *arXiv preprint arXiv:0809.1906*, 2008.
- [89] P. Owczarek y P. Zwierzykowski, "Review of simulators for wireless mesh networks," *Journal of Telecommunications and Information technology*, num. 3, p. 82, 2014.



- [90] “gnu.org.” [En línea]. Disponible: <https://www.gnu.org/licenses/old-licenses/gpl-2.0.html>
- [91] G. Seguin, “Multi-core parallelism for ns-3 simulator,” *INRIA Sophia-Antipolis, Tech. Rep*, vol. 106, p. 110, 2009.
- [92] W. R. Heinzelman, A. Chandrakasan, y H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *System sciences, 2000. Proceedings of the 33rd annual Hawaii international conference on*. IEEE, 2000, pp. 10–pp.
- [93] O. Borůvka, “About a certain minimal problem,” *Praca Moravske Prirodovedecke Spolecnosti*, vol. 3, pp. 37–58, 1926.
- [94] —, “Contribution to the solution of a problem of economical construction of electrical networks,” *Elektronický Obzor*, vol. 15, pp. 153–154, 1926.
- [95] J. Nešetřil, E. Milková, y H. Nešetřilová, “Otakar borůvka on minimum spanning tree problem translation of both the 1926 papers, comments, history,” *Discrete mathematics*, vol. 233, num. 1-3, pp. 3–36, 2001.
- [96] R. L. Graham y P. Hell, “On the history of the minimum spanning tree problem,” *Annals of the History of Computing*, vol. 7, num. 1, pp. 43–57, 1985.
- [97] C. F. Bazlamaçcı y K. S. Hindi, “Minimum-weight spanning tree algorithms a survey and empirical study,” *Computers & Operations Research*, vol. 28, num. 8, pp. 767–785, 2001.
- [98] F. Maffioli, “Complexity of optimum undirected tree problems: a survey of recent results,” in *Analysis and design of algorithms in combinatorial optimization*. Springer, 1981, pp. 107–128.
- [99] A. Pierce, “Bibliography on algorithms for shortest path, shortest spanning tree, and related circuit routing problems 1956-1974,” *Networks*, vol. 5, num. 2, pp. 129–149, 1975.
- [100] C. A. Hidalgo, B. Klinger, A.-L. Barabási, y R. Hausmann, “The product space conditions the development of nations,” *Science*, vol. 317, num. 5837, pp. 482–487, 2007.
- [101] P. Hagmann, L. Cammoun, X. Gigandet, R. Meuli, C. J. Honey, V. J. Wedeen, y O. Sporns, “Mapping the structural core of human cerebral cortex,” *PLoS biology*, vol. 6, num. 7, p. e159, 2008.
- [102] Network visualization - brain connectivity toolbox. [En línea]. Disponible: <https://sites.google.com/site/bctnet/visualization>
- [103] M. Everett y S. P. Borgatti, “Ego network betweenness,” *Social networks*, vol. 27, num. 1, pp. 31–38, 2005.